FUJITSU PERSONAL COMPUTER MAGAZINE パソコン情報誌

Oh!FM

季刊 FUJITSU MICRO 7 8 11

第2号 '83 spring

480円

本格派パソコンFM-11の全容

FM-7ハードウェア回路図と解説

FLEX® 入門/フロッピーディスクユニット

ゲームリスト
- さがしものゲーム
- 川下り
- ウェディングレース

- ●6809マシン語入門
- ●高速3Dグラフィックス

信頼と創造の富士通

FM-7 MAZE

¥126,000（本体価格 簡易言語ソフト付）

FM-7

セブン

興奮パソコン。新発売

青少年は興奮する。
新しい感性を伝えてくれるパソコン〈FM-7〉。マニアのために、新登場。
富士通
富士通株式会社 ■半導体統轄営業部 〒105 東京都港区虎ノ門2-3-13 TEL(03)502-0161
■札幌営業所(011)271-4311 ■東北営業所(0222)64-2131 ■長野営業所(0262)26-8222 ■静岡営業所(0542)54-9131
■名古屋営業所(052)201-8611 ■大阪営業所(06)344-1101 ■広島営業所(082)221-2288 ■九州営業所(092)411-6311
マイコンスカイラブ●東京・虎ノ門(03)591-1091/591-2561●東京・秋葉原(03)251-1448/251-1449
●大阪(06)344-7628/341-0486●名古屋(052)221-6016

3機種そろって興奮させる

青少年は興奮する。

高性能のパソコンがより手軽に、よりパーソナルに、より身近に使いこなせるようになりました。FM-8の先進技術を活かした豊富な最新機能と、多彩なソフトウェアを駆使できる魅力を携えて、FM-7、エキサイティングに新登場。性能で、価格で青少年を興奮させます。

まずご家庭のカラーテレビに接続して楽しめます。FM-7は、手持ちのカラーテレビを使って楽しむことからスタートできます(アダプタが必要)。カラーグラフィックや、市販のゲーム、ホビーのソフトを走らせるなど、楽しみ方はいろいろ。もっと解像度が欲しい方には、高解像度カラーCRTディスプレイも用意しています。

FMシリーズのF-BASICを標準実装。盛り沢山の機能をサポートしています。32KBをマスクROM化し、本体内に標準実装。これは、機能強化したFM-8用F-BASICをベースにしたもので、プログラミングが容易です。FM-8用に流通しているソフトウェアがそのまま使えるほか、フロッピィディスクベースのOSも、UCSD Pascal、FLEXCP/M-80を用意。FM-8や各種パソコンに流通しているアプリケーションプログラムの中から、目的に合ったソフトウェアを選べ、ホビーからビジネスまで幅広く対応できます。

カラーグラフィック機能も充実しました。
画面表示文字数は最大80字×25行(2,000字)、家庭用テレビでは40字×20行(800字)が最適です。3画面(3ページ)の表示内容を瞬時に切り換えてディスプレイできるマルチページ機能も付いています。カラーグラフィック画面は640×200ドットの高分解能表示が可能。1ドット毎に8色まで色指定ができ、文字との混在、文字の拡大もできるので、より美しく細かく多彩な表示が行えます。表示されている色をドット毎に瞬時に変化させるパレット機能も持っており、工夫しだいで楽しみ方は、さらに拡がります。

サウンドクルージングが楽しめるサウンド機能を標準実装しています。本体にシンセサイザ用LSIを内蔵。三重和音までのメロディや効果音を出すことができます。パソコンによる作曲や演奏が楽しめる機能です。画面と音が同時に出せるので、ゲーム効果音もバッチリ。外部スピーカとの接続もでき、ボリュームコントロールも可能です。ミュージックからゲームまで、コンピュータ・サウンド・クルージングが楽しめます。

日本語表示で読みやすく、ワープロとしても使いこなせます。オプションの漢字ROMカードを本体に装着することにより、FM-8用の日本語ワープロ・ソフトが使えます。JIS第1水準の漢字(2,965種)を、16×16ドットで画面表示。さらに印字することもできるので、友人にパソコン・レターを出すなど、身近なところで有効に使えます。

豊富な周辺装置を揃え、入門用から実務までFM-7があればOKです。数多くの周辺装置をFMシリーズとして揃えています。ディスプレイはグリーン/カラーCRT、フロッピィディスクは薄型1ドライブ/2ドライブ。プリンタもドットプリンタ/シリアルドットプリンタを用意。用途に合せてシステム構成できます。また、I/O拡張ユニットも接続でき、入門用から実務レベルまで、応用範囲はさらに拡がります。

簡易言語ソフトを標準装備しています。
FM-7は、家計簿や住所録など各種データ作成に便利な作表計算プログラムを標準装備しているので、購入したその日から役立てることができます。

先端技術が夢中にさせる 興奮パソコン。新発売

FM-7

¥126,000(本体価格 簡易言語ソフト付)

信頼と創造の富士通

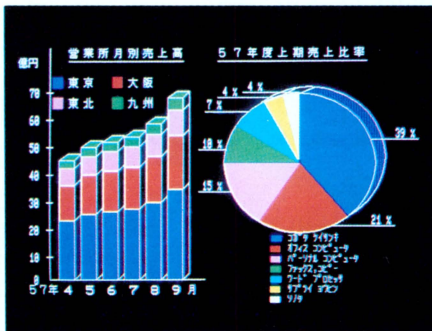
富士通のパソコン。FMシリーズ。

ディスク搭載の16ビットパソコン。

オフコンにせまる本格派。EDP部門から現場まで本格的ビジネスユースに対応できます。
FMシリーズの高級機種として、FM-11が新登場。EXタイプはCP/M-86を標準装備。これによりCIS-COBOL(ANSI-74準拠)、Pascalなどの高水準言語を利用できます。そして、BASICは、使いなれたFM-8と相互互換性のあるF-BASIC。オフコンでのデータ処理を分散化したいというEDP部門と、今まで使っていたFM-8のBASICでデータ処理したいという現場双方で、同一機器の使用が可能になりました。これにより社内のコンピュータシステムを統轄するEDP部門が、専門家としての立場からセクションマシンをサポートできるわけです。

フロッピィディスクを標準装備、そして分離型。操作性、扱いやすさをさらに向上。FM-11は、本体、キーボード分離のセパレート・タイプ。本体とキーボードとの接続コネクタは、本体裏側から。しかもケーブル(カールコード)は2.5mの長さです。オフィスのさまざまな条件、環境に合せて自在に設置でき、使いやすさに加え、ケーブルもキチンとキーボードに整理できます。キーボードは操作性を重視したロープロファイル、シリンドリカル・ステップ・スカルプチャー設計。FM-11 EX、ADの本体にはフロッピィディスクが1ドライブ標準装備され、さらに1ドライブ増設可能です。

日本語処理機能をいっそう充実。日本語処理をサポートしたCP/M-86の採用により、COBOLなどの高水準言語でも日本語処理が行えます。もちろんF-BASICでも処理可能。カーソル位置に漢字表示でき、漢字の拡大、外字登録も可能、表示文字数は最大40字×25行、などなど、使いやすさは抜群です。(漢字ROMカード、オプション)

メモリは128KBを標準実装、最大1MBまで増設可能。メインメモリとして、128KB標準実装されており、さらに本体内の拡張スロットに増設RAMを実装すれば最大1MBまで拡張できます。これによりCP/M-86、MS-DOS、OS-9などのOSのもとで、大容量メモリを必要とする高水準言語を余裕をもって利用できます。

多彩な機能を発揮するCRT表示。最大640×400ドットのグラフィック表示は、キャラクタとの混在表示も可能。カラー表示は16色。最大12画面を瞬時に切り換えて表示できるマルチページ機能、色交換が簡単にできるパレット機能、上下さらに左右へも可能になったスクロール機能、などCRT表示機能がより多彩になりました。

目的に応じて選択できる周辺装置。8インチ標準フロッピィディスク(1MB×2/台。2台まで接続可)は、ホストマシンとのデータ交換などに。大量データ収集には10MB(UNFORMAT時)のハードディスク。さらに、環境に左右されず扱いやすい大容量の128KBバブルカセットなど、用途目的に合せて周辺装置を選べます。

簡易言語ソフトを標準装備。FM-11EX、ADには簡易言語として作表計算プログラムFMCALCを標準装備、導入したその日から実務に役立てることができます。

ニーズに合せてお求めやすい3タイプ。

- EXは、薄型フロッピィ1ドライブを標準装備。CP/M-86、F-BASIC、FMCALCを添付。
- ADは、薄型フロッピィ1ドライブを標準装備。F-BASIC、FMCALCを添付。
- STは、F-BASIC(ROMバージョン)。

*FLEX、UCSD Pascal、OS-9、CP/M-80・CP/M-86、MS-DOSはそれぞれTSC社、カリフォルニア大学理事会、マイクロウェア社、ディジタル・リサーチ社、マイクロソフト社の登録商標です。

ビジネス用途を大きく拡げる高級パソコン。 新発売

FM-11

イレブン

■FM-11 EX ¥398,000(本体価格 簡易言語ソフト付)
■FM-11 AD ¥338,000(本体価格 簡易言語ソフト付)
■FM-11 ST ¥268,000(本体価格)

高級ホビーからビジネスまでの多才パソコン。

エイト

¥218,000(本体価格)

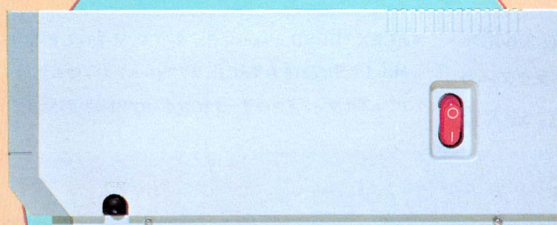
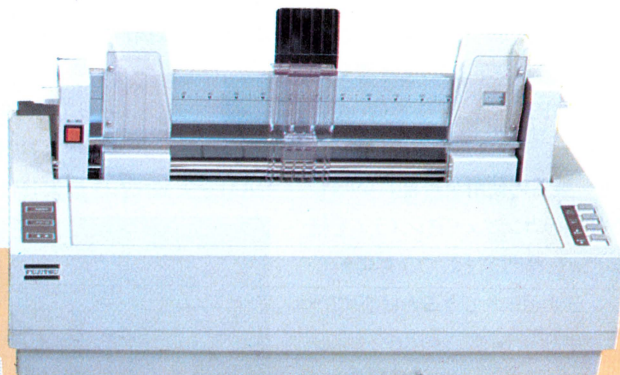
富士通

富士通株式会社

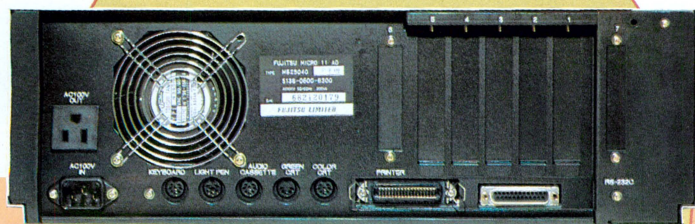
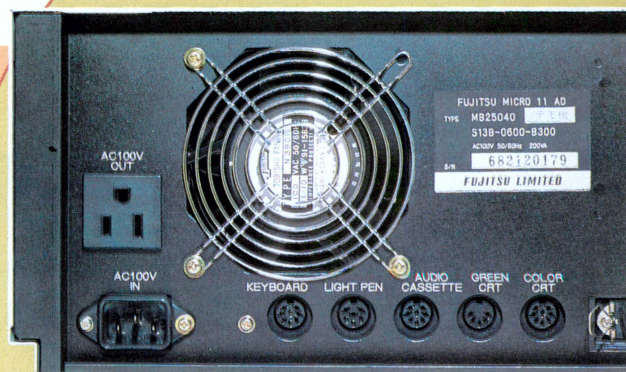
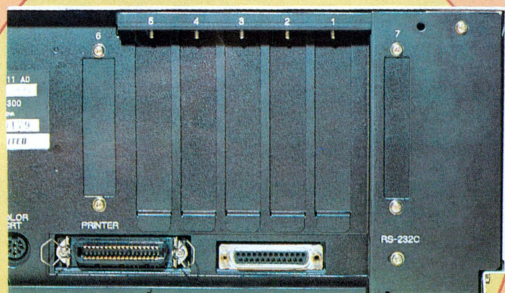
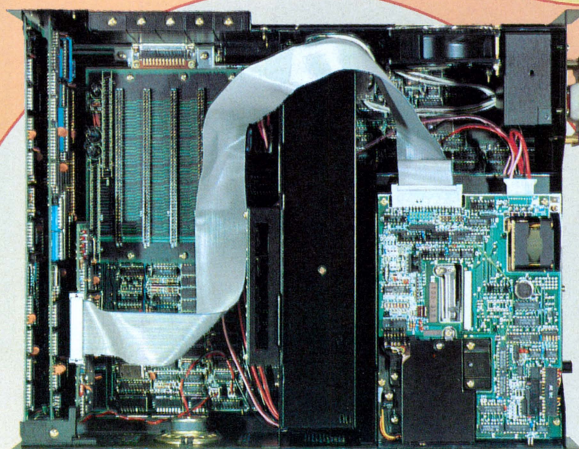
■半導体統轄営業部 〒105 東京都港区虎ノ門2-3-13 TEL (03)502-0161
●札幌営業所(011)271-4311 ●東北営業所(0222)64-2131 ●長野営業所(0262)26-8222 ●静岡営業所(0542)54-9131 ●名古屋営業所(052)201-8611 ●大阪営業所(06)344-1101 ●広島営業所(082)221-2288 ●九州営業所(092)411-6311
マイコンスカイラブ●東京・虎ノ門(03)591-1091/591-2561●東京・秋葉原(03)251-1448/251-1449●大阪(06)344-7628/341-0486●名古屋(052)221-6016

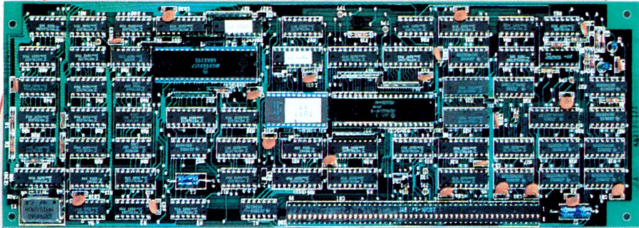
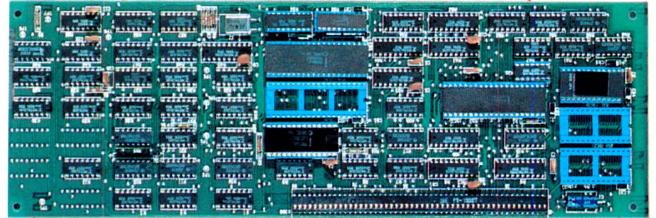
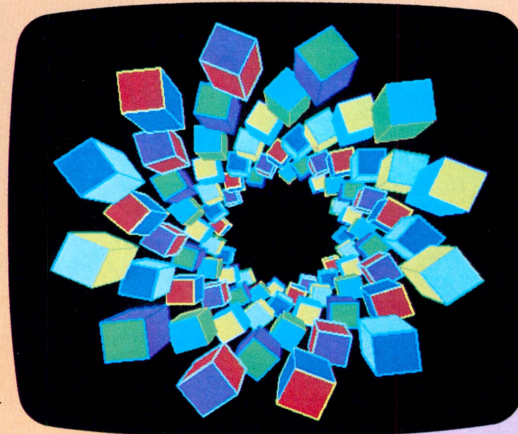
PHOTOGRAPH

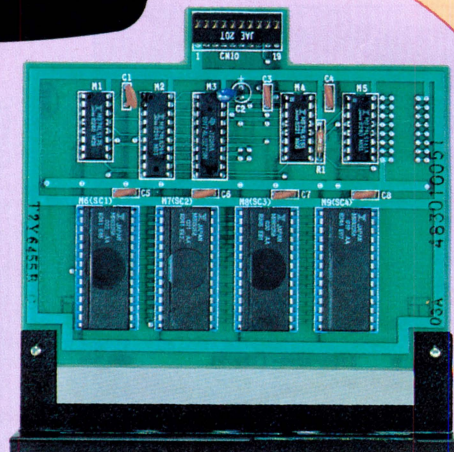
FM-11本体に，専用カラーディスプレイ（16色表示に対応），ビジネスプリンタを接続したところ

本体正面。本体のみではディスクドライブは1台だが，写真のように本体内に2台まで実装できる。下部のふたを開けるとDIPスイッチが見える。

本体側面。左下の黒いボタンがリセットスイッチ，右の赤いスイッチは電源である。

キーボード部。FM-7, 8とは，カーソルキー，テンキーの位置が異なる。

REPORT FM-11

本体裏面。各種コネクタおよびファンが見える。

本体裏面左側。モニタ用の100V出力は本体スイッチと連動している。キーボードのコネクタが背面に，またライトペンコネクタも見える。

本体裏面右側。プリンタ及びRS-232Cのコネクタのほか，種々のインタフェースボード実装時のためのスペースがある。

カバーをはずしたところ。本体に差されているボードは，左から6809CPUボード，8088カード，フロッピーコントロールカードが実装されている。右に見えるのはディスクドライブ。ケース背面と中央にファンが見えている。

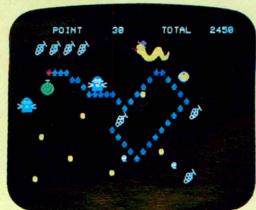
PHOTOGRAPH REPORT FM-11

6809CPUボード。メインCPUのほか，DMACなどが実装されている。

16ビットCPU 8088カード。このカードを実装すれば，CP/M-86™，MS-DOS™などの16ビット用ソフトが使える。演算プロセッサ，増設ROM用のソケットがついている。

FM-11の3Dグラフィックの一例。種々の大きさの立方体を表示。

本文18ページのSYMBOL@実行例のカラー写真。

漢字ROMカード。FM-7と同じ256KのROMを使っている。

さすがFMのグラフィックス……。

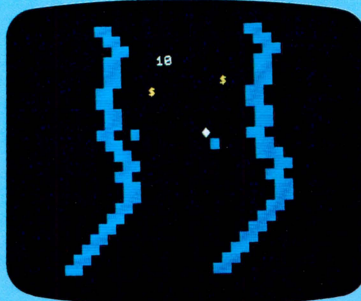
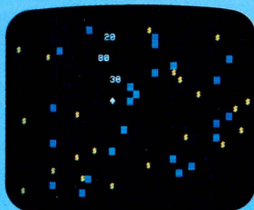
さがしものゲーム

（本文67ページ）

メロンを２個捜します。モグラを見つけるとアウト。♠が追いかけてくる。うまく逃げきるんだ！

ワープして……やっと♠から逃げ出せた，と思ったらあっ，モグラを出してしまった。

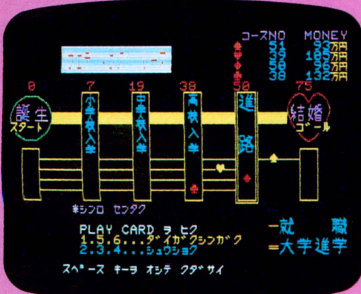
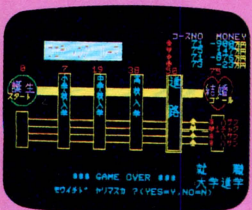
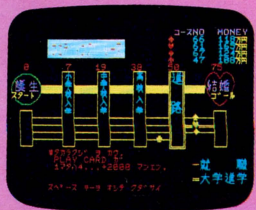
得点が悪いとこういうメッセージがでます。

川下り

（本文73ページ）

◇をうまく動かして■（障害物）をよける。■にぶつかったら沈没しちゃうよ。＄をとると10点から90点の得点がつく。

やっとゴールが見えてきた。一安心して気をゆるめちゃいけない。まだ■はたくさんある。

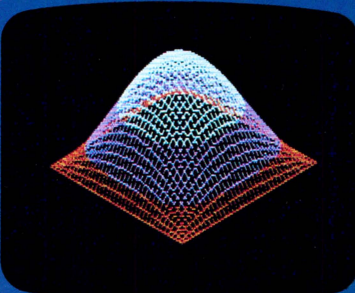
２画面クリアするとボーナスチャレンジができる。＄をとるとスコアに加算される。頑張るんだ!!

ウェディングレース

（本文76ページ）

◆の人は進路選択!! 進学するか就職するか，運命の別れ道だ。♣の人はまだ高校入学，ＰＬＡＹ ＣＡＲＤをうまくひこう。

やった，チャンス到来!! ここでいっきょに2000万円手に入るぞ！

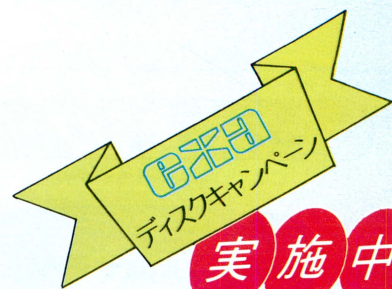
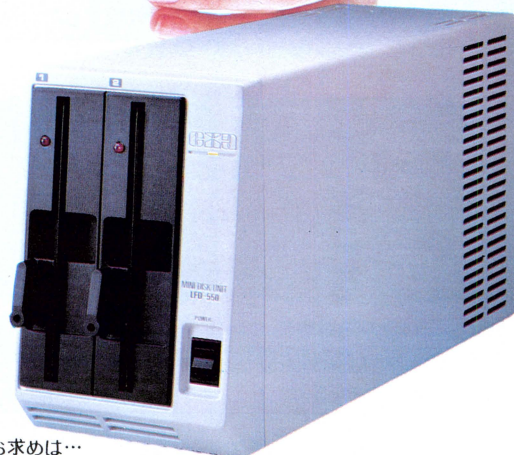
ゲーム終了。♥以外はみんな破産。お金は大事に使おうね。

高速3Dグラフィックス

本文106ページ

スピードをお見せできなくて残念。あっという間にできあがります。

タイプ練習プログラム

本文33ページ

練習４日目。だんだん難しくなってきた。

exa
5¼のラブレター
exa ディスクキャンペーン 実施中!
●アンケートを6/30までに送っていただいた方にもれなくTシャツプレゼント!
●さらにお買い上げのお店にてオリジナルポスターを差し上げます。
→5/31
ディスケットもexa。
LFD-550FM
Mini Disk Unit•Dual Type FUJITSU FM-7/8専用
¥148,000 汎用・増設用 ¥128,000
記憶容量●655,36KB/UNIT●327.68KB/ドライブ ●80トラック/ドライブ ●16セクタ/トラック●256B/セクタ ※2台標準実装(4台まで増設可)
ユーザーオプション●I/Oポート,ケーブルセット…¥15,000●富士通DISK BASICまたは各種OSディスケットが必要です。
exa 総発売元
東京電子科学機材株式会社
〒101東京都千代田区外神田2-2-12福井ビル☎03-257-1361(代)システム販売部
〒556大阪市浪速区日本橋東1-12-4☎06-631-1900(代)大阪営業所
〒980宮城県仙台市堤通り雨宮町3-16-201 ☎0222-72-8783(代)仙台営業所
＊お求めは…
お近くの電気店，マイコンショップ，デパート，スーパーでどうぞ。

ビジネスマン・OLのための本格派教室です。

ビジネスマン・OLのための本格派教室です。

と逃げるのか「やってみよう」と乗り出すのか？

いま“マイコン武装”を

●いま、日本のビジネス社会では、文字通り「秒単位」のスピードでマイコン革命が進行しています。マイコンの正しい知識と正確な技術を伴った“マイコン武装”は、運転免許と同様、これからのビジネスマン、OLにとって絶対に欠かすことができません。この現実から「逃げる」のか、それとも「挑戦する」のか――その意志決定こそ、あなたがマイコン時代に生き残れるかどうかのバロメーターになるのです。

全国どこの教室でもハイ・レベルな均一カリキュラムを

●ウチダマイコンスクールは、全国どこの教室でも質の高い均一なカリキュラムを受講できます。特に、企業では今春入社のフレッシュマンへの社員教育にご活用いただければ、さらに効果的です。

社員教育の一環に「いつでも」「どこでも」……一度ご相談下さい。

ビジネスユース、パーソナルユースに合わせて選べる、多彩なソフト パソコンさらに、パワーアップ——。

パソコンの概念を変える数々の機能を、高電社はさまざまなオリジナルソフトで実現してきました。これは、ソフト開発に定評ある当社ならではの快挙。これからも、パソコンをいっそう有効にご活用いただくためビジネスフィールドを中心に、ニーズに応えるソフトウェアをどんどん開発していきます。ご期待ください。

簡易言語シリーズ

簡易言語 SERIES

情報検索型文字データベース

PARAM/K1 漢字

FM-8 •PC-8800 •マイブレーン3000……… **¥49,000**

1. 項目(データ名)の数と長さ、画面、プリンター出力が自由設定できます。
2. 並べかえ、追加、修正、削除は簡単。
3. 1件(1レコード)64文字から128文字まで。
4. 複合条件(AND、OR、NOT)で検索します。
5. 1行は漢字仕様で53文字。
6. 複合条件(例えば東京都、男性、25才以上、未婚)で必要なデータを検索して表示印刷します。

パソコンは日常語で! プログラミングは自由自在!!

PARAM/1·2·3

パラム1(情報検索型)——使用機種FM8·PC-8000·PC8800 ¥39,000
パラム2(縦横計算型)——使用機種FM8·PC-8000·PC8800 ¥39,000
パラム3(マトリックスグラフ作成型)——使用機種FM8·PC-8000·PC8800 ¥39,000

ワードプロセッサー・ソフトシリーズ

マイレター 〔日本語ワードプロセッサー〕 漢字

FM-8 •PC-8800 •マイブレーン3000……… **¥49,000**

(カナ漢字変換方式)
1. 使用文字種 漢字JIS第1水準2965文字、非漢字453文字。
2. ディスプレイ表示文字数 40字×20行(10行)。
3. 単語事典20,000語登録 (オプション¥20,000)
4. 訂正・挿入・削除は簡単。
5. アンダーライン、センターリング、可能。
6. 禁則処理有 1文章=40×100行。
●縦書き、倍角文字可能。

英文ワードプロセッサー〔ワード9000〕

FM-8 •PC-8800 •マイブレーン3000……… **¥33,000**

医学用ソフト(大阪大学医学部マイコンクラブ監修)

RIA-MATE 〔ラジオイムノアッセイデータ処理プログラム〕

FM-8 •PC-8800 •PC-9800 •PC-8000… **¥70,000**

1. 通常の検査室でホルモンの測定等に使われるRIAのCompetitive methodに対しては、logit-log 2次多項式を用いて回帰しており、様々な測定物質に対し良好な回帰が得られる。
2. IRIやAFPの測定及び免疫関係の研究室でもよく使用されるRIAのnon-Competitive method (sandwich method)に対しては、4係数logistic曲線を用いて回帰しており、良好な回帰が得られている。
エンザイムイムノアッセイ(EIA)にも使用可能である。
3. 標準曲線のグラフが得られプリンターに出力可能である。
4. 標準曲線の不適合度を調べて、回帰が良くないときは、飛び離れ点を省いて計算しなおすことが出来る。

ビジネスソフトシリーズ

BUSINESS SOFT

漢字人名簿 〔DM宛名書可能〕

FM-8 •PC-8800……… **¥49,000**

1. 本システムは名刺整理、顧客資料、会員など多数の人名を登録しておき、必要な資料、宛名等を即座に作成します。
2. 1枚のフロッピーで2200名迄登録、検索、追加、修正、削除、並べかえができ、一覧表や宛名印刷ができます。

カタカナ仕様ビジネスソフトシリーズ

ESCO2000 〔見積実行予算システム〕

FM-8 •PC-8800 •PC-8000……… **¥90,000**

1. 提出用見積書、原価見積書が同時に作成できます。
2. 資材、工賃、諸経費等の分類・集計を迅速に行います。
3. 実行予算の項目指定は自由、予算の作成は簡単に行います。
4. 実績の消化状想は随時、ワンタッチで見られます。
5. 実績の明細は2000行まで記憶しています。

高電社ビジネスパソコンフェア

●4月1日(金)~4月30日(土)
●パソコン特別セール実施中!!

株式会社 高電社
マイクロプロジャパン一次代理店

マイコンショップ:システムイン高電社
〒546 大阪市東住吉区杭全7−10−15 TEL(06)719-1131(代)
大阪駅前店:システムイン高電社
〒530 大阪市北区梅田1-11-4大阪駅前第4ビル6F TEL(06)341-3371(代)

プログラム自由時代！

FM-7 8 11 用

プログラマーズ"パル"

```
***  ﾌﾟﾛｸﾞﾗﾑ ﾒｲ : DDDD  ***
ﾍﾝｽｳﾒｲ・ﾌﾞﾝﾊﾞﾝｺﾞｳ ﾀｲｵｳﾋｮｳ                      DATE : 00/00/00

ﾍﾝｽｳﾒｲ          :          ﾌﾞﾝﾊﾞﾝｺﾞｳ

  A
    140  1500  1520  2000  2010  2010  3000  3020  3020

  A$
    120   120   120   120   130   130   130

  C
    140   140   140   140  1500  1500  1500  1500  1520  1520  1520  1520  2000
   2010  3000  3030  3030

  D
   2000  2000  2020

  I
   2010  2010  3000  3020  3020  3020  3020  3020

  R
    140   140   140   140   140   140  1500  1500  1500  1500  1500  1500  1520
   1520  1520  1520  1520  1520  2000  2000  2000  2000  2010  2010  2020  3000
   3020  3020  3030

  T
   1000

  X
    140   140  1500  1500  1520  1520  2010  2010

  X(*)
   3020  3030  3030  3030  3030  3030  3030

  Y
    140   140  1500  1500  1520  1520  2010  2010

  Y(*)
   3020  3030  3030  3030  3030  3030  3030
```

```
  ***   ﾌﾟﾛｸﾞﾗﾑﾒｲ : << DDDD >>   ***
 ﾘｽﾄ 1                     DATE : 00/00/00

 100    REM          ======= TEST <<  DDDD   >>     ========
 110    WIDTH 80 , 25
     :  LOCATE 6 , 2
     :  PRINT " TEST PROGRAM  " ;
     :  LOCATE 4 , 5
     :  PRINT "1. ｳｽﾞﾏｷ" ;
     :  LOCATE 4 , 7
     :  PRINT "2. ﾁｮｸｾﾝ ｶｲﾃﾝ" ;
     :  LOCATE 4 , 9
     :  PRINT "3. ｾｲﾎｳｹｲ ｶｲﾃﾝ" ;
 120    LOCATE 4 , 16
     :  INPUT "SELECT NO. ==>>  " ; A$
     :  IF NOT (A$ = "1" OR A$ = "2" OR A$ = "3") THEN 120
        ELSE
          CLS
 130    IF A$ = "1" THEN
           GO SUB 1000
     :     GO TO 110
        ELSE
          IF A$ = "2" THEN
             GO SUB 2000
     :       GO TO 110
          ELSE
            IF A$ = "3" THEN
               GO SUB 3000
     :         GO TO 110
 140    A = 90
     :  FOR C = 0 TO 6
     :    FOR R = 0 TO 360
     :      X = 320+(C*36+R/10)*COS(R*3.14159/180)
     :      Y = 99+(C*12+R/30)*SIN(R*3.14159/180)
     :      PSET (X , Y , 7)
     :  NEXT R , C
```

```
ｻﾌﾞﾙｰﾁﾝ     :   ﾌﾞﾝﾊﾞﾝｺﾞｳ

   110 : Goto
    130   130   130

   120 : Goto
    120

  1000 : Gosub
    130

  1500 : Gosub
   1000

  2000 : Gosub
    130

  3000 : Gosub
    130
```

機 能

①1.構造化リスト
　2.強化構造化リスト
　3.変数番号対応表(変数リスト)
　4.サブルーチン 文番号対応表(クロスリファレンスリスト)

②ディスクバージョン(DISK BASIC用)
　テープバージョン(ROM BASIC用)

③出力　画面又はプリンタ

④使用ファイル　非アスキーセーブ(通常のセーブ)のプログラム。

⑤マシン語により高速処理実現
　構造化リストの場合、通常のリストとほぼ同速度
　変数リストの場合、20KBのプログラムで20秒(プリント時間を除く)

⑥変数文番号対応表、サブルーチン対応表では、特定の変数
　サブルーチンを指定することが可能。

⑦分割方式によるため、変数名の個数に制限はない。

価格		
	ディスクバージョン	￥18,000
	テープバージョン	￥15,000

C.C.W PACKAGES

等高線作画￥70,000
ＸＹ一平面に分布する任意量測定又は計算結果の等高線(等値線)を描く。

グラフパッケージ￥35,000
ファイルデータの受け渡しにより12種類のグラフを作成する。

F-BASIC入門 ￥25,000
類似問題の反復学習方式で間違えた問題は繰返し学習できる。

ファイル入門￥18,000
ファイルを作るための練習プログラムで即、実用にも使えるディスクを持っていても十分に生かせない人のために。プログラム説明書つき。

三次元座標基本ルーチン ￥45,000
2ベクトルのなす角、2平面の交線と交角、座標交換と点の回転、多点入力による球、球と平面の交線等17種類のルーチンから成っている。

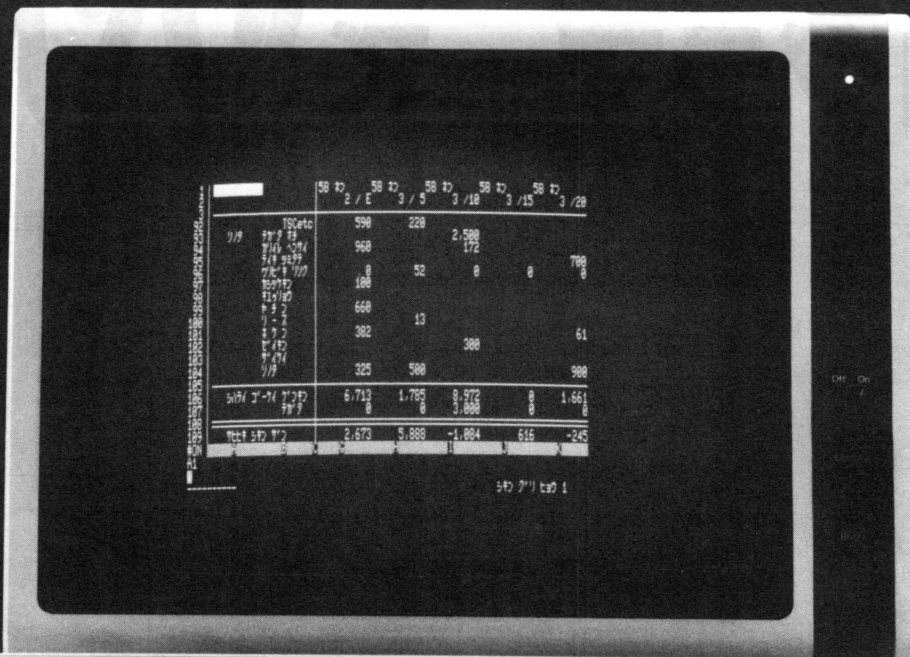
Condor S-20
リレーショナルデータベース￥260,000
CP/M､で稼動する本格派データベース。簡単なコマンドで専門プログラマーを必要としない。

ご注文は現金書留又は銀行振込でどうぞ
(銀行振込の場合は、はがきであらかじめご注文下さい)
○振込先　大和銀行新宿支店普通6108825

●お問合せは

株式会社　シー・シー・ダブル

〒151 東京都渋谷区代々木2-15-12-601 (03) 379-1831

ソフトマート簡易言語アプリケーション

FMCALC

FUJITSU MICRO 7・8用

ディスクバージョン…………………………￥68,000
カセットバージョン…………………………￥15,000

簡潔な命令で必要十分な処理を！

集計・ソート作業に威力を発揮するFM-8用本格的ビジネスソフトFMCALC、信頼のソフトマートがお届けします。

ワークシート方式で集計・ソート(アイウエオ順等の分類・並べかえ)などのわずらわしい業務を簡単なオペレーションで処理することができるようになります。FM-7、FM-8のハードウェアを効率的に使用することと、できるだけ少ない命令語で目的の処理を行なうことに工夫がこらされています。大企業の各セクションや、中小企業の経営等に威力を発揮する高速・高能力ビジネスソフトです。

主な特長

- ●POWER ONでFMCALCが起動します。他のソフト(FLEX、BASICなど)は不必要です。
- ●たいへん見やすい画面構成になっています。メッセージもできるだけ日本語です。
- ●数値精度は16桁、関数精度は13桁と、高精度の試算を高速で実行します。
- ●FM-7、FM-8のハードウェアと6809CPUの特性をフルに使用した機械語で書かれています。

FLEX™

FUJITSU MICRO 7・8・11

- ●FM-8用FLEX(5インチ or 8インチ)…………………￥58,000
- ●FM-7用FLEX(5インチ)……………………………￥48,000
- ●FM-11用FLEX(5インチ)…………………………￥48,000

※FLEXはTSC社の登録商標です。

①フレキシブルなDOS
標準でプリントスプーリング(プリントを出しながら他のことができる)を持ち、プリンタ、ターミナル等をフレキシブルに管理します。

②強力なFMS
ファイル・マネジメント・システム(FMS)は、非常に強力です。1セクター(256byte)ごとにディスクを管理し、ディレクトリー数の制限はありません。ユーザールーチンからのシステムコールが27種もあり、1度にいくつでもファイルをオープンできます。

③使いやすいシステムコール
文字の入力、出力、16進の出力などのシステムコールが23種あります。DOSもユーザープログラムからコールが可能です。

④多機能なアセンブラ・エディタ
標準でマクロアセンブラ、ラインエディタが含まれます。アセンブラは6800の命令を6809の機械語に落とすクロス能力の他、多くの機能があります。エディタもインサートサーチ、チェンジ、ムーブ、オーバレイ… 等々のコマンドを持ち、テキストの大きさに制限がありません。

⑤豊富なユーティリティ
デバッガー、クロスアセンブラ、高級言語をはじめ多くのユーティリティがオプションでそろっています。

SMart
SOFT MART, inc.

TSC社日本総代理店・スーパーソフト社 JAPAN AUTHORIZED DISTRIBUTOR

ソフトマート株式会社

〒101 東京都千代田区神田須田町1-18-6 第1谷ビル ☎(03)-256-5881

■取扱店

INTERRUPT東京	☎(03)256-6325
INTERRUPT横浜	☎(045)312-2325
INTERRUPT大阪	☎(06)245-7575
INTERRUPT福岡	☎(092)671-2466

目　次

季刊'83春号
富士通パソコン情報誌
Oh! FM
第2号

■お知らせ■

Oh! FMは第3号より
隔月刊誌
になります。
第3号は5月18日発売です。

本格派パソコン FM-11の全容

慶応大学メモリーマイコンクラブ
長沼 孝仁

FM-11の概要

FM-7と同時発売されたFM-11は，従来の8ビットのパソコンに行き詰まりを感じていたユーザーにとって，実に興味深いものであろう。その構成は8ビットから16ビットへ，大容量メインメモリ，高速(?)補助記憶装置が当たり前になりつつある現在のパソコン状況を如実に反映している。8ビットから16ビットへの過渡期にFM-8の上位機種として設計されたマシンの全容を紹介すべく，ここでは概要をお知らせしよう。

FM-11の三大特質

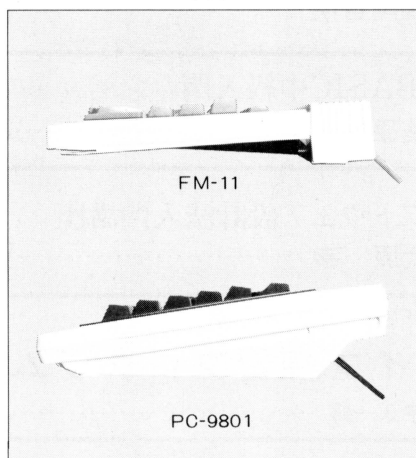
①作表言語が標準添付

FM-CALCという作表言語がSTを除くバージョンで，付属している。後述する富士通の販売戦略からのことであろう。このFM-CALCは付属といっても，ディスク版の高級作表言語であり，11購入後比較的早い時期からの使用が可能となる。

②キーボード分離，ディスク内蔵

本体とキーボードが分離される，高級パソコンとして現在ポピュラーな形態をとっている。本体とキーボードを接続するコードはPCシリーズと比べ軽く，移動性・可搬性が良い。

これは，キーボード制御用の4ビットマイコンをキーボード側に内蔵させ，本体にはデコード済みの信号を送るために，信号線が最低限で済むためである。またPCシリーズのキーボードと比べ薄型となり，操作性が向上していると思える。PCシリーズのキーボードとの比較を**写真1**に示す。

写真1

キータッチは軽く，ストロークが浅くなっており，FM-7のようにカチャカチャといった音はせず，FM-8のように指が沈んでいく感じもしない。ただ本体とキーボードとのコネクタが本体背面にあるのが気になる。またキー配列が変わり，筆者にはやや使いにくい配列となった。このカーソル移動キー，どうにかならないだろうか。

ディスク内蔵型をとった11は，①の点と合わせオフィスでの使用を十分考えているようである。実用的な使用では，本体と他の周辺は別々になっているよりも内蔵の形で組み合わせられているのが望ましい。ただ，それを進めすぎれば，システムの拡張性・任意性が減少してしまう。現在オフィスではもちろん，趣味においてもディスクが必需品となっているため，このようなタイプが適当と言えるのではないだろうか。以上のような点から実践的なマシンとしての思想が製品全体に貫かれているようである。

③豊富なドキュメンテーション

さすが，大型機のトップメーカーである富士通は，ドキュメンテーションを重要視している。マニュアル類は読みやすく，かなり詳しい。パソコンの中では，FMシリーズが最もマニュアルの整備において，優れていると言えるだろう。

特徴

表1に特徴を挙げておく。

FM-11には3つのバージョン(EXcellent, ADvanced, STandard）があり後者の方が廉価普及版である。その違いを**表2**に示す。違いは8088とディスクドライブが実装され

ているか否かにあり，ＳＴではともになし，ＡＤではディスクのみあるということになる。

表１　FM-11の特徴

① CPU
メインCPUとして 8088，Z80，68B09E が可能。
サブCPUは68B09
従来のFMシリーズと同様にメインCPUとI/O処理用のCPUが独立

② メモリ
メインRAMは128KB標準実装，最大1MB。VRAMは192KB標準実装

③ キーボード
本体と独立，非常に薄い

④ テキスト画面
テキスト画面とグラフィック画面が独立。専用CRT使用でカラー16色

⑤ グラフィック画面
640×400カラー２画面，640×200モノクロ12画面。専用CRT使用で16色中8色表示

⑥ ミニフロッピーディスク
内蔵可能。EX，ADには１ドライブ実装済

⑦ カセットインタフェース
標準実装。1600bps，半速サッポロ方式

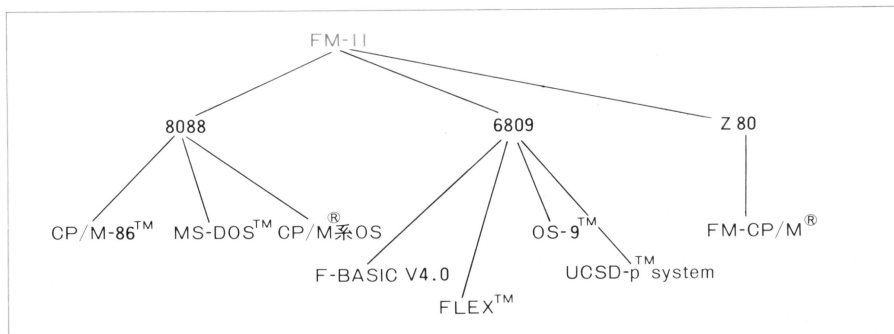
⑧ ライトペン
インタフェース内蔵。分解能80×25

⑨ 外形，重量
本　　体 464×360×153mm 11.8kg
キーボード 464×193×39.5mm 1.7kg

⑩ 価格
ＥＸ　¥398,000
ＡＤ　¥338,000
ＳＴ　¥268,000

表２　EX，AD，ST比較表
○は標準実装
×はオプション

	ＥＸ	ＡＤ	ＳＴ
8088カード	○	×	×
ミニフロッピーディスク1台同インタフェース	○	○	×
CP/M-86	○	×	×
F-BASIC(ディスク)	○	○	×
F-BASIC(ROM)	×	×	○
FM-CALC	○	○	×

表３　CPUとOS

これからのパソコンはディスクがなければ使えないと考える筆者としては，ＳＴは必要ないと思えるのであるが……。

FM-8と同様，このマシンでもデュアルＣＰＵ方式を取り，I/Oプロセッサを別に持っている。I/Oプロセッサ（サブＣＰＵ）は68B09で変えられないが，メインＣＰＵは，16ビットを含む三つのＣＰＵを使うことができ，様々なＯＳソフトに対応可能である。その様子を**表３**に示す。

この方式によって既存のＯＳ上で走る様々なユーティリティソフトを走らせることができ，また，8088があるために将来性をも持った息の長いマシンと言えそうである。しかし，これらのＯＳ上のソフトはほとんどユーティリティであり，現在パーソナルユースで非常に大きなウェイトを占めるアミューズメントソフトは全くないと言ってよい。

今までのところパソコンはアミューズメントソフトにより普及し，その後ユーティリティなど充実してくるというパターンが多く，ユーティリティマシンとして作られたパソコンの中には，普及せず消え去ったものも少なくない。FM-11においてもFM-8のゲームソフトが走るよう考えてもらいたいところである。

メインCPUのメモリは128ＫＢが標準実装されており，この状態のときF-BASICモードでフリーエリアが76ＫＢほどある。サブＣＰＵはＶＲＡＭに192ＫＢのＲＡＭを持つ。

この192ＫＢという大規模なＶＲＡＭによって，640×400のカラー高解像度画面を二つ持つことが可能になり，640×200のモノクロ画面ならば12画面持てるようになった。この12画面を切り換えて使えば，簡単なリアルタイムアニメーションも可能である。グラフィック画面はパレット機能を持ち，専用ＣＲＴを用いれば16色中８色の表示ができる。この16色というのは従来の８色にそれぞれ高輝度色（赤→ピンク，黒→灰，青→水色……など）が増えたもので，このため一般のＣＲＴは使えない。一般のＣＲＴでは輝度信号のＲＧＢ入力がデジタル化してあり，高輝度色が表現できないからである。

表４　周辺装置

○カード
8088（CP/M-86付）　¥80,000
Z-80（FM-CP/M付）　¥45,000
漢字ROM(ワープロ付)　¥46,000
ミニフロッピーディスクコントロール　¥20,000
標準フロッピーディスクコントロール　¥30,000
ハードディスクインタフェース　¥14,000
拡張RAM　¥58,000(128KB)　¥88,000(256KB)

○装置
高解像度カラーCRT　¥188,000
高解像度グリーンCRT　¥49,800
ミニフロッピーディスクドライブ　¥65,000
標準フロッピーディスクドライブ　¥350,000
ハードディスクドライブ　¥498,000
ライトペン　¥48,000

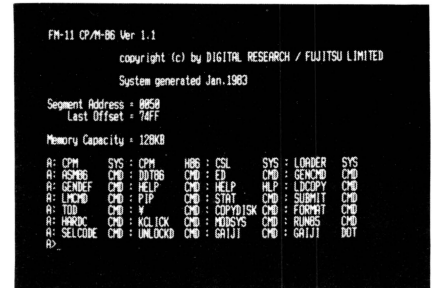
テキスト画面は，横80・40桁，縦20・25行の各モードが可能で，グラフィック画面と独立している。また，８とのコンパチビリティを考え，テキスト画面とグラフィック画面が同時にスクロールするモードも用意されている。テキスト画面のスクロールが非常に高速化している。ＣＲＴＣ（6845）の使用によるためか，ＰＣ88などより速く，ファンクションキーを表示している状態でもかなりのスピードでスクロールする。従来のＦＭを知っている者にとっては非常にうれしいことだ。

表４に周辺装置の一覧を示す。プリンタなど他のＦＭ用のものを使用可能な場合は除い

てある。FM-11ではライトペンインタフェースが内蔵されている（ＣＲＴＣの機能をそのまま使用している？）ためにライトペンの使用が容易である。

この表からわかるとおり，カードにはソフトが付属している。富士通は原則的にハードには最低限のソフトを付けて売ることにしているようだ。一見ユーザーにとっても有利のような売られ方だが，種々の問題をはらんでいる。ソフトの重要性が認識されたためにこのような販売方式を採ったのだろうが，逆に「ソフトはハードのおまけ」とも取れる。本題からそれるのでここでやめておくが，今後の動向に注目されたい。

ハード，ソフトの詳細については別稿で取り上げる。概略について以上で紹介を終わったわけだが，興味深いマシンであることがおわかりいただけただろうか。ハードに関しては現在のパソコンのトップにあると言える。メモリ・ＣＰＵ・グラフィックスなど，今パソコンに要求されている機能に，現時点で最高のテクニックを持って近づこうとしている。が，今いち使いにくい。使用にあたってはソフトが大きくものを言うためである。僕には英語がわからない。そんな人間には海の向こうのＯＳが何種類のっていてもうれしくないのである。

ＯＳが何種類も使える点は非常に大きな特徴である。将来性を考えると，パソコンでは最も期待できるマシンであろう。今後のソフトの充実を切に望むものである。

CP/M-86

FM-11のソフト

第１部で紹介したとおり，ＦＭ-11では３種類のＣＰＵがメインＣＰＵとして使用でき，よってそれぞれのＣＰＵ上で様々なＯＳが走る。最初にＭＢＬ68Ｂ09をメインＣＰＵとしたときに動作するF-BASIC V4.0について詳しく見てみよう。

F-BASIC V4.0

ＦＭ-11では３種のバリエーションとも，6809を実装しているために，即F-BASIC V4.0を使用できる。ただし，ＳＴにはディスクが付属していないため，F-BASIC V4.0はＲＯＭで供給される。

図１はメモリマップである。F-BASIC V4.0はこの広いメモリ空間を管理する。ＢＡＳＩＣインタプリンタは図のBASICコード領域に位置している。BASICのテキストは図のプログラムテキスト領域に書き込まれ，ＰＣ-8801と同様テキストウィンドウを通してアクセスされる。この図でユーザＲＡＭ領域が変数エリアであり，BASICの変数エリアとテキストエリアが別々であることがわかる。

F-BASIC V4.0では，変数名を40文字まで識別するようである。が，先頭に予約語を含むとエラーとなる。ここは，N98や88と同様，先頭が予約語の変数名も可とし，予約語はスペースで区切って使うようにした方が良いかと思えるのであるが。

ＦＭ-11ではBIOS，サブシステムとも改良されており，特にサブシステムの改良点が多いと言える。これに関しては後述するが，サブシステムの改良，機能強化に伴いF-BASICにも変化が見られる。これを**表５**に示す。FM-7，8ではキーバッファが１文字分であったのが大幅に増加しているのがわかる。

図１ メモリマップ
メインRAM（128K Bytes）

アドレス	領域
0	BASIC RAM領域
0600	ユーザーRAM領域
7C00	テキストウインドウ
8000	BASICコード領域
FC00	I/O等
FFFF	

アドレス	領域
10000	プログラムテキスト領域
13000	BASICコード領域
17C00	プログラムテキスト領域
1FFFF	

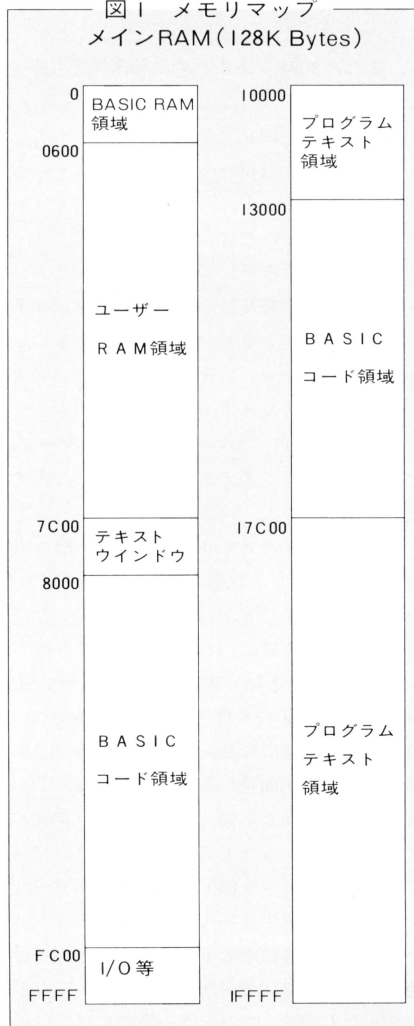



表５ F-BASIC操作機能の変更

- ○キーバッファが32文字。あふれは無視
- ○ファンクションキーがリピートされる
- ○ブレークキーではキーバッファが消去されない（従来と同じ）
- ○ＥＳＣコードが使用可能
- ○ＣＬＳキーの強化
- ○ローカルエディットモードの確立

ファンクションキーがリピートされるようになった。どのキーもオートリピートするということで統一されたようだが，誤操作が増えそうな気がする。誤ってＬＩＳＴ(ファンクションキーの２）を押し過ぎた場合，ブレークキーを押してもキーバッファがクリアされないから，[ESC]+[9]でキーバッファをクリアしてからブレークキーで止めなければならない。

N98やN88BASICでは，グラフィック画面を消去するのに CLS 2 を実行せねばならず，消去に手間がかかった。11では[SHIFT]+[CLS]キーでグラフィック画面が，[CTRL]+[CLS]キーでテキストとグラフィックの両画面が消去され，非常に便利である。11ではグラフィック画面の消去が非常に速い点も特筆に価する。[SHIFT]+[CLS]キーは一見，ＣＬＳ５と等価であるようだが，少し違っている。

ローカルエディットモードが11では，はっきりしている。従来のＦＭでは[INS]キーの打鍵によってインサートモードが終了したが，11では[Return]，[CTRL]+[C]，[CTRL]+[X]のローカルエディットモード終了を示すキーに

よっても終了する。つまりリターンキーを押せばインサートが終了するのである。

表6にエスケープコードの詳細を示す。いろいろとおもしろいコードがあるが，スクロールスピードを変更するために[ESC]+[X]+[ ]がある。11のテキストスクロールが速すぎるため，そのスピードの可変に用意されたのだろうが，[X]キーの次の一文字のアスキーコードが大きいとスクロールが遅くなってしまう。電源ONの状態でスクロールスピードは最高である。

V4.0の新設命令

表7の新設命令について詳しく述べたい。

グラフィック・漢字関係では4つのコマンドが新設された。

ROLL命令はグラフィック画面のスクロールを行う。対象は画面全体であり，はみ出た部分は消去され新しく現れる画面はクリアされている。この命令では縦方向のスクロールのほか，横方向のスクロールも可能となっている。が，横方向には16×*n*ドットのスクロールしかできず，1ドットずつなめらかに行うことができない。これはCRTCの機能を利用したためと考えられる。

FM-8や7では漢字にシンボル文が用意されておらず，不満を感じることが多かったが，V4.0ではSYMBOL＠命令で可能となった。

11ではユーザーによる漢字キャラクタフォントの定義が可能である。ユーザーが定義可能な文字は960文字で，DEF KANJI文によって定義する。定義方法は，ドットパターンを16進数列で表現する，文字列で表現する，の2とおりがある。

KANJI文は，漢字を連続的に書くとき便利なコマンドである。このコマンドによってテキスト画面の文字とグラフィック画面の漢字を，混在して書くのが容易となった。

実行例1，2にそれぞれDEF KANJI・SYMBOL＠のデモ，KANJI文のデモを示しておく。

11ではコマンド1つで画面に時計を表示でき，BASICのプログラムが走っていなくても表示を続ける。これがCLOCK ON命令で，テキスト画面の任意の位置へ時計の表示を行

表6　エスケープシーケンス機能(キーボード)

キー		機能
ESC	1	カーソル位置にタブを設定します。
ESC	2	カーソル位置のタブを解除します。
ESC	3	すべてのタブ設定を解除します。
ESC	9	キーバッファに入っているキーコードをクリアします。
ESC	*	テキスト画面を消去します。
ESC	T	1行のうち，カーソル以降を消去します。
ESC	Y	カーソル以降の画面を消去します。
ESC	E	1行挿入します。
ESC	R	1行削除します。
ESC	X (*n*)	スクロールの速度を指定します。*n*は0～9の任意の値を入れます。数が大きくなるほどスピードが遅くなります。
ESC	g	バッファリング機能を有効にします。
ESC	h	バッファリング機能を無効にします。

表7　F-BASIC　V4.0-V3.0比較表

分類	命令・関数	内容
○新しく追加された命令		
漢字・グラフィック関係		
	ROLL	グラフィック画面のスクロール
	KANJI	漢字表示命令
	DEF KANJI	ユーザーによる漢字定義
	SYMBOL@	漢字のシンボル文
テキスト画面関係		
	COLOR@	テキスト画面のアトリビュート変更
	CLOCK ON/OFF	テキスト画面への時計表示
ライトペン関係		
	PEN ON/OFF/STOP	ライトペン割り込みの設定
	ON PEN GOSUB	ライトペン割り込みの処理
その他		
	WRITE	ファイル出力に便利なPRINT文
	STOP ON/OFF	ストップキー(ブレークキー)による中断の許可
○機能が変更になった命令		
グラフィック関係		
	SCREEN	パラメータ増加，変更
	CLS	パラメータ増加
	COLOR	パラメータ増加
	LINE	ラインスタイル可能
	PAINT	タイリング可能
テキスト画面関係		
	LOCATE	カーソルの形変更可
	WIDTH	桁数指定0も可(論理的桁数無限大)
	CONSOLE	カラースイッチなし
音楽関係		
	PLAY	単音のみ
	SOUND	単音の高さと長さ指定
その他		
	CLEAR	スタック領域指定可
	AUTO	機能強化
	HARDC	領域指定可
	FILES	横に並ぶ
	TERM	機能強化
○新しく追加された関数		
	PEN	ライトペンステータス関数
○機能が変更になった関数		
	FRE	パラメータ増加
	SCREEN	読み取るアトリビュート増加
	数値関数	倍精度演算可能

実行例 1

実行例 2

う。CLOCK OFFは表示の停止命令である。表示は残念ながら針式ではない。

WRITE文はPRINT文とほぼ同様の命令であるが，数値を出力する際，符号の位置のスペースは出力されず，また文字列は前後を"（ダブルクォーテーション）で囲まれ出力される。

STOP ON/OFF命令は，ブレークキーの有効無効を指定する。つまり，STOP OFF命令によってブレークキーが無効になり，プログラムの実行を中断できなくなるのである。無限ループになった状態でSTOP OFF されていようものなら，中断することはできず，Hang upに落ち入ってしまう。使用にあたっては十分気をつける必要がある。

PEN ON/OFF/STOP, ON PEN GOSUB命令は，見てのとおり，ライトペン制御用命令である。キーやタイマの割り込み制御命令と同様の使い方をする。

V4.0での機能変更命令

グラフィック関係

11ではVRAMとして192KBもの RAMが標準実装され，640×400のカラー画面を２画面，640×200の単色画面ならば12画面を持つことができる。この大規模なＶＲＡＭのモードを決定するのが，SCREEN命令である。SCREEN命令について**表８**に示す。ここでおもしろいのは，ビデオ出力コードである。これを使えば白黒モニタでプログラムを表示しながら，カラーモニタにグラフィックを表示することも可能となる。が，専用高解像度カラーモニタと一般のモノクロモニタが同時に使えるのだろうか？

ＣＬＳ命令は，画面モードが増えたためにパラメータが変更となった。ＣＯＬＯＲ命令でもパラメータが増え，グラフィック用のフォアグラウンドカラー，アトリビュートが追加された。アトリビュートの指定にインテンシティというのがあり，これを使えば専用モニタ使用時に16色表示が可能となる。

SHIFT＋CLS キーがＣＬＳ ５という命令と似ているが，実は違うと先に述べた。どちらもグラフィック画面の消去に使うのだが，COLOR命令によってバックグラウンドカラーを変更したときに違いが出るのである。ＣＬＳ命令によって画面を消去すると，そのときのバックグラウンドカラーで消去するのであるが，SHIFT＋CLS では違うようなのだ。

ＬＩＮＥ命令では，N88などと同じようにラインスタイルの指定ができる。ラインスタイルとは，線のパターンのことで，この指定によって点線や鎖線が表現できることになる。

ラインスタイルと同時にPAINT命令でのタイリングも可能となった。やはりこの機能もN88などが持っていたもので，基本8色の組み合わせで様々な色を表現するものである。

表８　SCREEN

〔機　能〕

グラフィック画面の画面モードやページの設定などをおこないます。

〔形　式〕

SCREEN　〔画面モード〕〔,〔アクティブページ〕〔,〔ディスプレイページ〕
〔,〔アクティブバンクコード〕〔,〔ディスプレイバンクコード〕
〔,ビデオ出力コード〕〕〕〕〕

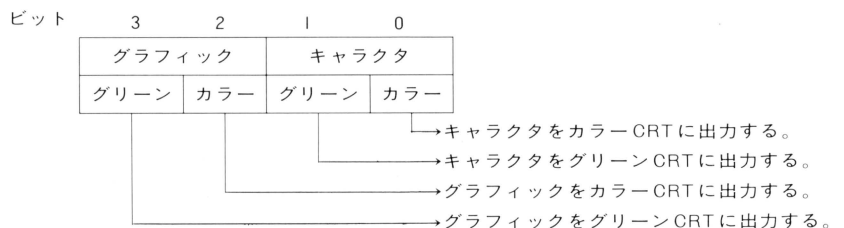
〔説　明〕

・画面モードは，グラフィック画面の解像度やカラー表示，単色表示，漢字表示のモードを設定します。パラメータは，０～５の値をとり，次のような意味になります。

〈画面モード〉

0――カラーモード（640×200ドット）
1――カラーモード（640×400ドット）
2――単色モード　（640×200ドット）
3――単色モード　（640×400ドット）
4――グラフィックスクロールモード（640×200ドット）
5――グラフィックスクロールモード（640×400ドット）

〈バンクコードとページ〉

○ビデオ出力コード

先述のラインスタイルもこのタイリングも表現方法（表記法）はN88と同じである。

テキスト画面関係

LOCATE命令によって，カーソルの形を変えることができる。普通はキャラクタサイズの長方形なのだが，指定によってアンダーラインにも，正方形にもできる。**表9**にカーソルの形について示す。

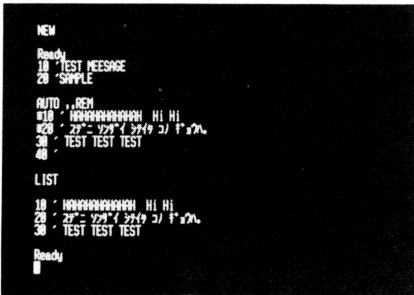
音楽関係

11はＰＳＧを持っていないために，ＦＭ-7のような豊富な音楽機能がない。

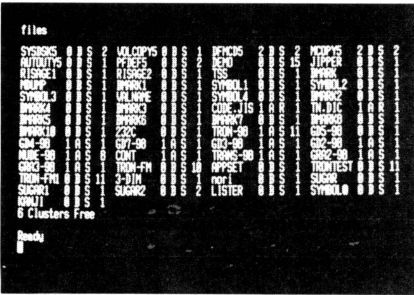
PLAY命令では単音しか出せないが，MML（音譜の書き方）は同じである。ＳＯＵＮＤ命令はFM-7の命令とは全く違い，鳴らす音の周波数と長さを指定できるに過ぎない。よって，この命令はＢＥＥＰ命令の変形と考えるのが妥当であろう。

その他の命令

AUTO命令が，かなり改良されている。N88同様，既に存在する行をAUTO命令によって発生すると，行番号の前にアスタリスク（＊）を表示する。AUTO命令のフォーマットを**表10**に示す。この表で3番めのパラメータにREMオプションがある。このオプションを付加すると，行番号の後ろにクォーテーション（'）を発

実行例 3

生する。この様子が**実行例3**に示されている。

ＦＭ-11のＡＵＴＯ命令では行番号の直後にリターンを押してもＡＵＴＯが終了せず，[BREAK]，[CTRL]＋[C]，[CTRL]＋[X]の打鍵によ

FIES実行命令

り終了する。また，ＡＵＴＯ中でもスクリーンエディットが可能で，既に入力した行を修正することができる。

ＣＬＥＡＲ命令では，N88，N98と同様に文字列領域の大きさを指定する第１オペランドは意味を持たない。11では新たに第３オペランドが追加され，BASICが使用するスタック領域を指定することができるようになった。デフォールト値は512（バイト）である。

ＨＡＲＤＣ命令では，コピーするエリアの

表 9　LOCATE

〔機　能〕　画面上の任意の位置にカーソルを移動します。
〔形　式〕

LOCATE　〔水平位置〕〔，〔垂直位置〕〔，〔カーソルスイッチ〕〔，〔カーソルスタート〕〔，カーソルエンド〕〕〕〕

カーソルスタートとカーソルエンドにより，カーソルの形状を指定します。指定できる値の範囲は，CRTディスプレイの種類と画面の行数により，右表の様になります。
なお，カーソルエンドの値はカーソルスタートの値より大きくなければなりません。

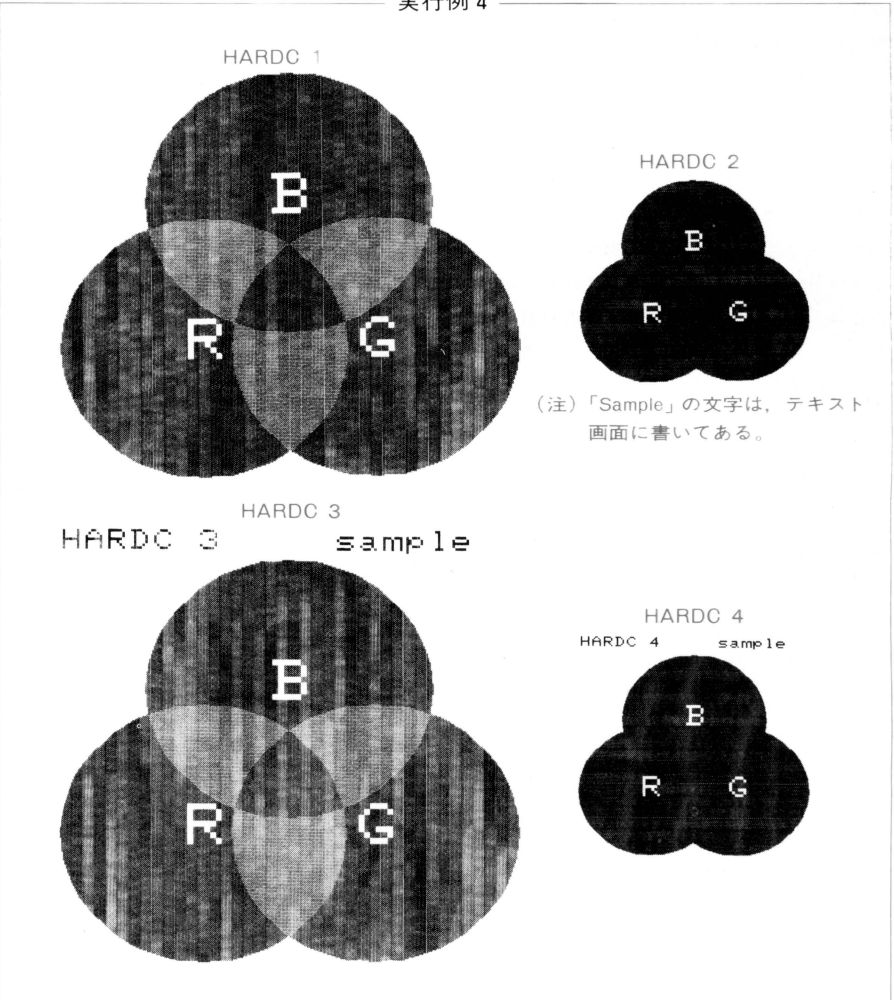
CRTの種類＼行数	25行	20行
200ライン	0 ～ 7	0 ～ 9
400ライン	0 ～15	0 ～19

表10　AUTO

〔機　能〕
行の先頭に行番号を自動的に発生します。
〔形　式〕

AUTO　〔行番号〕〔，〔増分値〕〔，REM〕〕

実行例 4

（注）「Sample」の文字は，テキスト画面に書いてある。

指定ができるようになり，またモードが増えている。1ではグラフィック画面のみを拡大コピーし，濃淡がつき，2ではグラフィック画面のみ縮小コピーし，濃淡はつかない。3, 4のモードはそれぞれ，1・2にテキスト画面を加えたものである。**実行例4**はHARDC命令の使用例である。

FM-7, 8ではFILESをとると，ファイル1つで1行を取るため，ファイルの一部が画面からはみ出してしまい，全てのファイルを1画面に出せないことが多かった。が，11では80文字画面では1行に4つのファイルが表示されるため，多くのファイルを1画面で見ることができるようになった。

TERM命令は大幅に機能を拡張している。

コントロールコードによる機能も増え，コントロールコードの意味も，F-BASICオーダとADM-3Aオーダの2種類が選べる。

またエスケープシーケンスが加わり，ホストから端末へ細かな命令を送ることが可能となった。エスケープシーケンスについては，**表6**に挙げたものプラスαとなるため，省略することとし，F-BASICオーダによるコントロールコード表を**表11**に示すことにする。

表11　F-BASICオーダ

内部コード（16進）	アスキーニーモニック	動　作・意　味
05	ENQ	現在のカーソル位置からそのフィールドの終りまでを消す。
07	BEL	ベルを鳴らす。
08	BS	カーソルを1つ左に移動する。
09	HT	次のタブ停止位置までスペースを発生する。
0A	LF	改行する。
0B	VT	カーソルをホームポジションへ移動する。
0C	FF	テキスト画面を消去する。
0D	CR	復帰する。
11	DC1	Xパラメータ指定のとき，XONコードに使用。
		Xパラメータ指定のないとき，フィールドの開始を指定。＊
12	DC2	カーソル位置を指定する。＊
13	DC3	Xパラメータ指定のないとき，XOFFコードに使用。
		Xパラメータ指定のないとき，文字を反復表示する。＊
15	NAK	グラフィック画面を消去する。
16	SYN	テキスト画面とグラフィック画面の両方を消去する。
1B	ESC	エスケープシーケンスの開始コードとして使用する。
1C	ES	カーソルを右に移動する。
1D	GS	カーソルを左に移動する。
1E	RS	カーソルを上に移動する。
1F	VS	カーソルを下に移動する。

＊印のついたコードは，複数バイトシーケンスで構成される。
11　DC1(at)　　at：アトリビュート
12　DC2(x)(y)　x,y：座標値
13　DC3(n)(c)　n：表示回数　c：文字
注：(　)は1文字のパラメタを示す。

このTERM命令を使って，慶応大学の計算センターに接続を試みた。が，画面が頻繁に消去され，使用できないのである。そこで画面にコントロールコードを表示させるモードとし（PF8によって，このモードになる）調べてみたところ，大型計算機から$16というコードが送られて来ていることがわかった。このコードはF-BASICオーダ・ADM3-Aオーダともにテキスト・グラフィック画面の消去を意味するのである。N88, 98では大丈夫だったのであるが。プリンタへエコーバックした例が**実行例5**である。

N88, 98ではTERMモードでのバッファが非常に大きく，そのバッファがロールキーによって簡単に参照することができる。V4.0でもぜひ持って欲しかった機能の1つである。

V4.0の関数の機能変更

PENはライトペンに関する情報を与える関数で，PEN(n)の形で用いる。nの値によって種々の情報が得られる。

FREは空きメモリの量を与える関数であり，FRE(n)の形で使用する。V3.0まではnが何であっても未使用領域の量を示したが，V4.0では，0で未使用変数領域，1で未使用テキスト領域，2で0と1の合計，文字列は0のときと同じ領域の量を示す。

SCREEN関数ではテキスト画面のアトリビュートが増えたために，読み取るアトリビュートもブリンク（点滅），インテンシティ（高輝度）などが増えた。

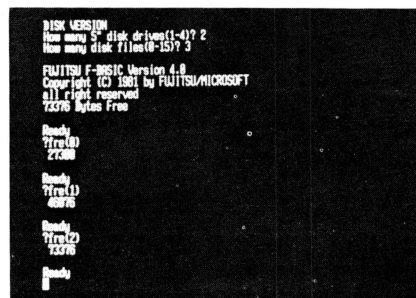
ベンチマークテスト

ベンチマークテストとは，あるプログラムを各機種で走らせ，処理スピードを計ろうとするものである。が，個々に特徴を持った命令が存在するため，比較が，特に時間による比較が難しい。この点を踏まえたうえで比較結果をご覧いただきたい。

各プログラムを説明する。なお，以下のプログラムはFM-11用である。走らなければ，パラメータなど機種によって最低限の変更を行っている。

1．数字を画面に出力する。改行は，せず続けて表示する。

2．数字を画面に出力する。表示の後，改行を行う。

電源ON後のフリーエリア

3．数値計算の例として，シンプソンの公式を使った定積分を行った。式は，

$$Y=\int_0^{\pi}\sin\theta d\theta$$

である。計算結果はいずれの機種でも2.00007となった。

4．文字列処理の例である。アスキーコードで32から255までの文字を並べ，その文字列を逆に，255から32へと並べ換えている。

5．LINE命令の例である。5のプログラムは縦が400ドットの画面を想定している。このような画面を持てるのは，PC98

実行例5　TERM(プリンタへのエコーバックを行ったもの)

```
JCB936I NO INTERPRET TABLE
JCB931I PLEASE LOGON
LOGON TSS H05850 T(,1)
EFLD26I H05850 TIME=(0000,01),REGION=0320K
ENTER PASSWORD FOR H05850-

H05850 LOGON IN PROGRESS AT 12:52:14 ON FEBRUARY 19, 1983
 << DATA PROCESSING CENTER KEIO UNIV - FACOM M180/2AD >>
 ***** PLEASE WAIT TO BE -- READY -- *****
READY
LISTC
READY
LISTC
IN CATALOG:SYSCTLG
SPECIFIED LEVEL NAME H05850 NOT INDEX STRUCTURE
LASTCC=4
READY
LOGOFF
********** TSS ACCOUNTING INFORMATION **********
* USER-ID << H05850 >>      AIFNO << 13064 >>    *
* START << 12:52:14     >>  DATE << 83-02-19 >> *
* END   << 12:53:51     >>                      *
* CPU   << 00:00:00:190>>   TPUT <<         6 >> *
* ELAPS << 00:01:37     >>  TGET <<         3 >> *
* CALCULATION CHARGE .......... <<        21 >> *
* REMAINDER ................... <<       710 >> *
************* TSS SERVICE END *****************
H05850 LOGGED OFF AT 12:54:03 ON FEBRUARY 19, 1983+
KEQ54100I SESSION ENDED
```

とFM-11だけなので,5′に,縦200ドットのモードを使って行ってみた。

6．主にPAINT命令のスピードを測定した。このプログラムも5と同様の理由から6′を用意し，各機種について測定した。

結果を表12，13に示す。この中でPC98だけ16ビットCPUを搭載し，11を含む他の機種は全て8ビットCPUを使用している。気付いたことを2,3挙げてみよう。

○TEST 1においては，98も11もほぼ同じスピードなのだが，TEST 2になると98が11より4倍も時間をとっている。これは，両機種のテキスト画面の違いからきているのである。PC98では,何とテキスト画面に漢字が書けるのである。それに比べFM-11では220種ほどの文字しか書くことができない。これが，スクロールスピードの差となって現れたのであろう。

○TEST 5において，98は11の10分の1になっている。これは98が，LINE命令をGDC(Graphic Display Controller)を用いて，ハードで行っているためであろう。

○TEST 6で，88のスピードが非常に遅いことに気づかれるだろう。確かに，1個のCPUで，48KBものVRAMを使うのは，きついことであろう。だが，それ以外にも重要な点がある。PC98や88では，WINDOW文やVIEW文があるのである。非常に便利な命令だが，LINEや,CIRCLE，PSETのときに，画面範囲を計算しながら実行しなければならない。11では，これを行っていないために，スピードの点で有利になっている。

個人的に言わせてもらえば，多少のスピードを犠牲にしても，WINDOW文を実現して欲しかったところだ。この機能があるとないでは，グラフィックス使用時にかなり大きな差がでる。

ベンチマークは，あくまで目安である。この点を理解して，この稿を役立てて頂きたい。様々な機種に，それぞれ特徴があり，この特徴を握むことができれば，かなりの性能を引き出すことが可能だからである。

F-BASIC V4.0 感想

ヒーローになろうとする者の前にはライバルが現れる，というのはB.C.3世紀アテネの哲人の言葉，ではもちろん，ない。人間界にはライバルなるものがあるのだが，パソコンでは，これに相当する対抗機種というのが存在している。

ほとんど正気の沙汰とは思えない乱造ぶりで有名なメーカーを除けば，富士通とNECという2大パソコンメーカーにおいて，機種の対抗関係は明確に現れている。富士通がFM-8を出せばNECがPC88を，FM-7を出せばMKIIという具合で，そこから考えれば当然，FM-11とPC98は対抗関係にあると言えるだろう。価格の面から考えても両機種は似たような購買層を持っていることだろう。

FM-11はすごい。ディスク内蔵型である。何より3つのCPUが走り，それぞれのCPU上のOS，それぞれのOS上のソフトウェア

テストプログラム 1

```
100 '
110 ' TEST PROGRAM 1 --- PRINT  No.1
120 '
130 WIDTH 80,25 : CONSOLE 0,25
140 DEFINT A-Z
150 '
160 TIME$="00:00:00"
170 FOR I=1 TO 20000
180   PRINT I;
190 NEXT
200 PRINT TIME$
210 '
220 BEEP
230 END
```

テストプログラム 2

```
100 '
110 ' TEST PROGRAM 2 --- PRINT  No.2
120 '
130 WIDTH 80,25 : CONSOLE 0,25
140 DEFINT A-Z
150 '
160 TIME$="00:00:00"
170 FOR I=1 TO 20000
180   PRINT I
190 NEXT
200 PRINT TIME$
210 '
220 BEEP
230 END
```

テストプログラム 3

```
100 '
110 '     TEST PRGRAM 3 --- CALCULATION CAPACITY
120 '
130 '    Integration by simpson method
140 '
150 DEFSNG A-Z
160 DEFFNY(X)=SIN(X)
170 X0=0 : X1=3.14159 : P=3000
180 '
190 TIME$="00:00:00"
200 H=(X1-X0)/2/P
210 D=0 :      X=X0  :  Y=FNY(X)
220 D=Y+D :    X=X+H : Y=FNY(X)
230 D=Y*4+D : X=X+H : Y=FNY(X)
240 D=Y+D : P=P-1
250 IF P>0 THEN 220
260 ANSWER=D*H/3
270 PRINT TIME$
280 '
290 PRINT "answer :";ANSWER : BEEP
300 END
```

テストプログラム 4

```
100 '
110 '  TEST PROGRAM 4 --- STRING PROCCESS
120 '
130 '   reverse character string
140 '
150 CLEAR 1000
160 DEFINT A-Z
170 '
180 TIME$="00:00:00"
190 FOR LOOP=1 TO 20
200   X$=""
210   FOR I=32 TO 255
220     X$=X$+CHR$(I)
230   NEXT
240   QI$=X$ : QO$=""
250   IF QI$="" THEN 280
260     QO$=LEFT$(QI$,1)+QO$ : QI$=MID$(QI$,2,LEN(QI$)-1)
270   GOTO 250
280   X$=QO$
290 NEXT
300 PRINT TIME$
310 '
320 PRINT X$ : BEEP
330 END
```

テストプログラム 5

```
100 '
110 '    TEST PROGRAM 5 --- GRAPHICS
120 '
130 '              LINE COMMAND
140 '
150 DEFINT A-Z
160 SCREEN 1
170 '
180 TIME$="00:00:00"
190 FOR I=0 TO 639
200   LINE (I,0)-(639-I,399),PSET,I MOD 8
210 NEXT
220 FOR I=0 TO 399
230   LINE (0,I)-(639,399-I),PSET,I MOD 8
240 NEXT
250 PRINT TIME$
260 '
270 BEEP
280 END
```

テストプログラム 6

```
100 '
110 '    TEST PROGRAM 6 --- GRAPHICS
120 '
130 '              LINE BOX AND PAINT
140 '
150 DEFINT A-Z
160 SCREEN 1
170 CLS 0
180 H=400
190 MAX=H/2 : YAXIS=H-1 : YCENT=H/2
200 '
210 TIME$="00:00:00"
220 FOR I=0 TO MAX STEP 5
230   COL=(I/5) MOD 8
240   LINE (I,I)-(639-I,YAXIS-I),PSET,COL,B
250   PAINT (320,YCENT),COL,COL
260 NEXT
270 PRINT TIME$
280 '
290 BEEP
300 END
```

が利用できる。サブシステムを持っているし，8ビットにしては高速である。それに我らユーザーの味方富士通は，I/OやBIOS，サブシステムなど全て公開してくれて，本体を買うとそれらの情報まで与えてくれるのである。何とドキュメンテーションのすばらしいことか。画面エリアに至っては，PC98の2倍，専用モニタを使えばカラー16色が表示できるのである。

以上のように非常にすばらしいFM-11であるが，ここではF-BASICについてPC98と比較してみよう。

感想が，感じ想ったことを述べることである以上，主観的になってしまうと思うが，お許し願いたい。

F-BASIC V4.0にあってN88(86)BASICにない命令もある。が，筆者が考えるかぎり，どう見ても98のBASICの方が，勝っている。V4.0にはラベルが使えない。WINDOW，VIEWが使えない。漢字のサポートが弱い。リストのローリング（リストを画面上で自由に上下に動かすこと）ができない。これらの点がプログラミングの際にどんなに重要か，経験のある方ならおわかりいただけるかと思う。

ラベルが使えないと見通しのよいプログラムが書きにくくなるし，コーディングも面倒なことになる。WINDOW，VIEWはグラフィック時には実に便利なコマンドで拡大，縮小も簡単にできるし，クリッピングが非常に楽である。3Dグラフィックスでクリッピングが楽というのは非常に有利な点なのである。漢字に関しては，FM-11でもKANJI文を用意し，使いやすくしたようだが，

KANJI　　&H3140;　　(V4.0)
PRINT　　"雲";　　(N88(86))

の2つでは，どう見ても下の方が見やすい。

エディットモードのローリングもデバッグ時には実に重宝するものである。重ねて言うが，以上の点はプログラムを書く身になって考えた，非常に主観的な感想である。

FM-11でもラベル付きのBASICは使える。OSがたくさん乗ることを利用してBASIC09™

表12　実 行 時 間

（´は画面を縦200ドットにしたプログラム）

	TEST 1	TEST 2	TEST 3	TEST 4	TEST 5	TEST 5′	TEST 6	TEST 6′
FM-11	1′31″	1′48″	1′54″	1′11″	2′02″	1′19″	1′45″	0′28″
FM-7	5′01″	24′09″	2′06″	1′23″		1′59″		0′34″
FM-8	10′30″	28′05″	3′31″	2′34″		3′16″		1′07″
PC98	1′27″	7′58″	1′40″	0′50″	0′13″	0′10″	6′38″	1′52″
PC88	4′45″	13′04″	3′46″	2′04″		3′31″		4′50″

表13　FM-11との実行時間比（FM-11を1とする）

	TEST 1	TEST 2	TEST 3	TEST 4	TEST 5	TEST 5′	TEST 6	TEST 6′
FM-7	3.31	13.4	1.11	1.17		1.51		1.21
FM-8	6.92	15.6	1.85	2.17		2.48		2.39
PC98	0.96	4.43	0.87	0.70	0.11	0.13	3.79	4.00
PC88	3.13	7.26	1.98	1.75		2.67		10.4

使用機器……FM-11：682120179, FM-7：K821100013, FM-8：381090694, PC98：2X002829A, PC88：K2059609（V1.0）

を走らせればよいのである。が，このBASIC 09，ディスクを必要とし，価格も結構なものなのだ。（11用の価格はわからない。8用のではOS含めて17万円ほどである）

以上述べたとおり，BASICのできには不満が多いというのが事実である。何とか富士通に頑張ってもらって，F-BASIC V5.0や8088用F-BASICの登場に期待したいものである。

FM-11 BIOS

FM-11マニュアルを買うと，FM-11が付いてくる。かなりオーバーな表現かな？

11のドキュメンテーションは非常にしっかりしていて，BIOSに関しては，「BIOS & ブートストラップローダ解説書」なる立派なマニュアルが本体に付属してくる。本稿では，これをもとにBIOSについて紹介していくことにする。

BIOS 概 要

F-BASIC V4.0のBIOSには，今までのRCB(Request Control Block)を持つエントリ方法のほかに，レジスタにパラメータをセットする方法も加えられた。

後者をレジスタインタフェースと言い，パラメータが少ないものに関しては，高速化するためにこの方法が使える。RCBを持つものについては，7，8と全く同じなので省略する。レジスタインタフェースの入出力については図2のとおりである。

BIOSはBASICと独立しているため，BASICから切り離して使うことができる。また，

表14　BIOSリクエスト番号

1. RCBインタフェース

リクエスト番号	名　　前	内　　容
0	ANALGP	アナログポート入力
1	REMOTE	オーディオカセットインタフェースモータコントロール
2	CTBWRT	オーディオカセットインタフェース出力
3	CTBRED	オーディオカセットインタフェース入力
4～7	リザーブ	
8	RESTR5	ミニフロッピーディスクリストア
9	WRTD5	ミニフロッピーディスクライト
10	REDD5	ミニフロッピーディスクリード
11	リザーブ	
12	BEEPON	ブザーON
13	BEEPOF	ブザーOFF
14	LPOUT	プリンタ出力
15	リザーブ	
16	SUBOUT	Display Sub System出力
17	SUBIN	Display Sub System入力
18	INPUT	コンソールキャラクタ入力
19	INPUTC	コンソールキャラクタ継続入力
20	OUTPUT	コンソールキャラクタ出力
21	KEYIN	1文字キー入力
22	KANJIR	漢字フォントの読込み
23	LPCHK	プリンタステータスチェック
24	BIINIT	BIOSイニシャライズ
25～27	リザーブ	
28	DRLOCK	標準フロッピーディスクドアロック
29	SETMOD	セクタ長指定
30	DISK2	標準フロッピーディスクメディアチェック
31	RESTR8	標準フロッピーディスクリストア
32	WRTS8	単密度標準フロッピーディスクライト
33	WRTD8	倍密度標準フロッピーディスクライト
34	REDS8	単密度標準フロッピーディスクリード
35	REDD8	倍密度標準フロッピーディスクリード
36	SEEKT5	ミニフロッピーディスクシークトラック
37	SEEKT8	標準フロッピーディスクシークトラック

2. レジスタインタフェース

リクエスト番号	名　　前	内　　容
0	ACHROT	コンソール1文字出力
1	AKEYIN	キーボード1文字入力
2	APRTOT	プリンタへの1文字出力
3	APRTE1	拡張プリンタ1への1文字出力
4	APRTE2	拡張プリンタ2への1文字出力
5	EXTRCB	RCBインタフェース拡張テーブルアドレスセット
6	EXTREG	レジスタインタフェース拡張テーブルアドレスセット
7	PRTCHK	プリンタステータスチェック
8	DMASEG	DMAセグメントの設定

ポジションインディペンデントであるため，どのロケーションでも実行することができる。

リクエスト番号の128から255（bit 7＝1）については，ユーザーが使用してBIOSを拡張することができるようになっている。

BIOSのエントリ方法

RCBを用いるときは7や8のときと全く同じで，IXレジスタにRCBのアドレスを入れて，

```
JSR 〔$FBFA〕
```

図2　BIOSレジスタインタフェース

〔BIOS CALLの時〕

Acc B	リクエスト番号
Acc A	8ビットパラメータ
IX	16ビットパラメータ

〔BIOSからのリターン時〕

Acc A	リターンパラメータ
Acc B	リターンパラメータ
Carry flag	1＝Error
〔$FF98〕	エラー内容

を実行する。この$FBFAは7，8と同じである。レジスタインタフェースの場合は，〔$FBFA〕＋3番地をサブルーチンコールする。マニュアルでは，あるメモリにエントリアドレスを作って，それを間接コールしなさいと書いてあるから，まず，初期設定中で，

```
LDD    $FBFA
ADDD   #3
STD    RENTRY
```

を実行してから，

```
JSR 〔RENTRY〕
```

とすればよい。

BIOSコマンド表

BIOSのコマンド表を表14に示しておく。

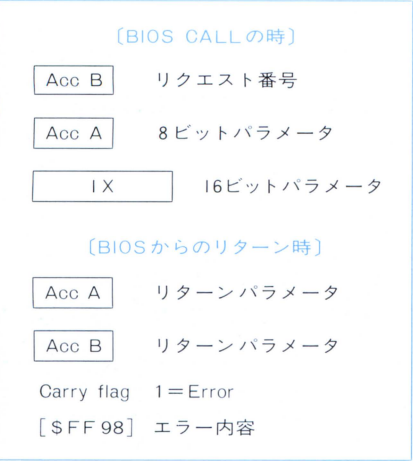



表15　サブシステムインタフェースアドレス

アドレス	R/W	内容	ビット内容
$FD04	リード	FIRQ内容	Bit 0：ATTENTION Bit 1：Breakキー
	ライト	Attention Acknowledge	Don't care
$FD05	リード	サブとのハンドシェーク	Bit 7：Busy
	ライト		Bit 6：Abort Bit 7：Halt

FM-11 Sub System

FM-11のサブシステムの使用法は，FM-7や8と全く同じである。サブシステムの詳しい使用法は，本稿の趣旨にそぐわないため，省略させていただく。

サブシステムの呼び方は，7，8と全く同じ，共有RAM空間にコマンドやデータを置いてインタフェースする。メインCPUが共有RAMをアクセスするときにサブCPUを停止させるのも，従来と同じである。

サブCPUとのハンドシェークに使うI/Oのアドレスを表15に，サブシステムのコマンドを表16に示す。

FM-11のハードウェア（6809での1MBアクセス）

6809はアドレスバスを16本，つまり64KBのアドレス空間しかないのだが，11では1MBをアクセスすることができるよう設計されている。6809ではMMU（Memory Management Unit）と呼ばれるLSIを使用すれば，1MBをアクセスできるが，11では使用されていない。が，ソフト上ではたいへんよく似た操作が実現されている。

表16　Display Sub System コマンド一覧表

1．コンソールコマンド

コマンド名	コード	機能	コマンド名	コード	機能
CONSOLE INIT	$01	コンソールパラメータの初期設定	TAB SET	$0B	タブ位置の設定
ERASE CONSOLE VRAM	$02	コンソールVRAMの消去	CONSOLE CONTROL	$0C	コンソール制御フラグの設定
PUT STRING	$03	文字列の出力	ERASE 2	$0D	WINDOW内のコンソール，グラフィックVRAMの消去
GET	$04	ローカルエディット動作の起動	CLEAR CONSOLE FLAG	$0E	コンソール制御フラグのクリア
GETC	$05	ローカルエディット後の変更フィールドの読取り	SET CONSOLE FLAG	$0F	コンソール制御フラグのセット
GET CHARACTER BLOCK 1	$06	VRAMからの文字コードの取込み	COPY CONSOLE VRAM	$10	コンソールVRAMの内容の複写
PUT CHARACTER BLOCK 1	$07	VRAMへの文字コードの出力	SELECT COMMAND END MODE	$11	コマンドエンド方法の選択
GET CHARACTER BLOCK 2	$08	VRAMからの文字，アトリビュートコードの取込み	CHARACTER OUT	$12	1文字表示
PUT CHARACTER BLOCK 2	$09	VRAMへの文字，アトリビュートコードの出力	GET LIGHT PEN POSITION	$13	ライトペン座標の読取り
READ CONSOLE PARAMETERS	$0A	カーソル位置	CHARACTER LINE	$20	文字で線を引く

2．グラフィックコマンド

コマンド名	コード	機　能
LINE	$ 15	線又は四角を描く
CHAIN	$ 16	点を連結した線を描く
POINT	$ 17	指定した点を描く
PAINT	$ 18	境界の内部をぬる
SYMBOL	$ 19	文字を拡大，回転して表示
CHANGE COLOR	$ 1A	枠内の色を任意の対応で変える
GET BLOCK 1	$ 1B	指定色のドットパターンを読取る
PUT BLOCK 1	$ 1C	指定色のドットパターンを表示
GET BLOCK 2	$ 1D	3原色のドットパターンを読取る
PUT BLOCK 2	$ 1E	3原色のドットパターンを表示
GRAPHIC CURSOR	$ 1F	指定座標の読取り
SELECT ACTIVE VRAM	$ 21	コマンドの対象となるVRAMの選択
SELECT DISPLAY VRAM	$ 22	表示の対象となるVRAMの選択
GET BLOCK 3	$ 23	指定色のグラフィックと文字パターンの読取り
READ DISPLAY STATUS	$ 24	グラフィックの動作状態の読取り
ERASE GRAPHIC VRAM	$ 25	グラフィックVRAMの消去
SET GRAPHIC START ADDRESS	$ 26	VRAMスタートアドレスの設定
KANJI SYMBOL	$ 27	漢字を拡大，回転して表示
READ FONT	$ 28	ASCII，漢字の文字フォントの読取り
COPY GRAPHIC VRAM	$ 33	グラフィックVRAMの複写
KANJI OUT	$ 34	コンソール座標で漢字を表示
DEFINE EXTERNAL FONT OF KANJI	$ 35	漢字外字フォントの定義
LINE 2	$ 36	線を描く（Circle Full用）

3．キーボードコマンド

コマンド名	コード	機　能
INKEY 1	$ 29	キーコード（8ビット）の読取り
DEFINE PF STRING	$ 2A	PFキーへ文字列を定義
GET PF STRING	$ 2B	PF定義文字列の読取
SET PF INTERRUPTION	$ 2C	PF割込の選択
INKEY 2	$ 2D	キーコード（9ビット）の読取り
ALLOCATE KEY BOARD	$ 2E	キーボードの配置の決定

4．タイマコマンド

コマンド名	コード	機　能
SET TIMER	$ 3D	タイマレジスタに値を設定する
READ TIMER	$ 3E	タイマレジスタの値の読取り
DISPLAY CLOCK	$ 40	時計表示機能の選択

5．継続コマンド

コマンド名	コード	機　能
CONTINUE	$ 64	データを継続して転送する

6．その他のコマンド

コマンド名	コード	機　能
TEST	$ 3F	絶対アドレスでの参照
CALL MACHINE	$ 7F	機械語ルーチンの起動

表17　メモリ管理レジスタ（MMRを除く）

アドレス	内　　容	R/W	データ							
			D7	D6	D5	D4	D3	D2	D1	D0
FD90	MMR SEGMENT REGISTER (MSR)	ライト	MMR SEGMENT NUMBER							
			*	*	*	*	S3	S2	S1	S0
FD91	WINDOW BANK REGISTER (WBR)	ライト	WINDOW BANK							
			*	*	*	*	A19	A18	A17	A16
FD92	WINDOW OFFSET REGISTER (WSR)	ライト	WINDOW OFFSET ADDRESS							
			OA15	OA14	OA13	OA12	OA11	OA10	OA9	OA8
FD93	MODE SELECT REGISTER (MSR)	リード/ライト	MMR 0：無効 1：有効 リセット時 0	WND 0：無効 1：有効 リセット時 0	*	*	*	*	(BASIC ROM) 0：有効 1：無効 リセット時 0	ブートROM 0：有効 1：無効 リセット時 0

メモリバンク

64KBのメモリが1単位となり，これをセグメントと呼ぶ。ただし，1セグメントは正確には63KBで，$FC00～$FFFFの範囲はバンクセレクトされない。この空間には，I/Oやワークエリアが置かれる。

11ではバンクセレクトやメモリのアロケートは4KB単位で行われる。この4KB単位のアロケートを行うのがMMR（Memory Mapping Register）で$FD80～$FD8Fに位置している。$FD80は6809の$0000～$0FFFが実際にどの部分に対応するかを決定し，たとえば，$FD80に2を書き込めば，$2000～$2FFFに対応し，$10を書き込めば，セグメント1の0000～$0FFF（つまり$10000～$10FFF）に対応することになる。

$FD90にMSR（MMR Segment Register）と呼ばれるレジスタがあり，MMRのセグメント部分を修飾する。セグメントのみのオフセットなので，下位4ビットのみ有効である。

ウィンドウ機能

図1（20ページ）にも示したとおり，$7C00～$7FFFがウィンドウと呼ばれ，他のメモリ空間を簡単に参照することのできる領域である。この領域に，どのメモリ空間を対応させるかはWBR（Window Bank Register）とWOR（Window Offset Register）により決定される。

話が元に戻るが，ウィンドウとは，他のメモリ空間と直接対応づけられる領域のことである。たとえば，ウィンドウに$10000～$103FFが対応しているとしよう。すると$7C00をアクセスすると，$10000をアクセスしたのと同じことになり，$7C00に書き込むと，$10000に書き込んだことになる。読み込みも同様で，$7C00を読めば$10000の内容が読み込まれる。

対応は，WBRとWORで決定される。WBRは，対応させる空間のセグメントを決定する。よって下位4ビットが有効である。WORはセグメント内での対応を決定する。$7C00と，実際のロケーションとのオフセットを持つのである。WORが0なら$7C00の内容は，そのまま$7C00に対応し，$10なら，$8C00を，$84なら$0000を対応させる。

表17に，MMRを除く主なメモリ管理レジスタを示す。

CP/M®，CP/M-86™はデジタルリサーチ社，MS-DOS™はマイクロソフト社，FLEX™はTSC社，OS-9™はマイクロウェア社，UCSD-P™はカリフォルニア大学理事会の登録商標です。

☆耳寄りな話☆

4月から富士通のTV番組「コンピュートないと」がスタート

4月4日から，富士通のFM-7，OASYSを使ったTV番組「コンピュートないと」がスタートする。テレビ東京およびテレビ大阪から，毎週月曜日P.M.11：15～11：45に放送される。

司会は連続テレビ小説でおなじみの相原友子，講師に東大・大型計算機センターの石田晴久教授を迎え，ゲストとのトーク，パソコン情報，なんでも相談室，ワープロ使用現場訪問などを盛り込んだバラエティ番組になる。

トーク＆講座では，毎回，様々なジャンルから著名人を招き，ゲストの得意な分野，関心事などをテーマにトークを展開，その内容に結びつけてパソコンの使い方を講義する。ゲストとテーマは，タモリ＝パソコン，ワープロの楽しさ，加山雄三＝星座表，ヨットシミュレーション，伊丹十三＝料理献立表，石田芳夫＝囲碁，阿佐田哲也＝カードで勝負，森英恵＝マイ・カラーデザイン……などが予定されている。

FMユーザーおよびFMファン，パソコンやワープロを知りたい者にとって，月曜の夜は楽しみな時間となりそうだ。

COMDEX/FALL '82に出品された FM−16

昨年11月29日～12月2日ラスベガスで，INTERFACE主催によるCOMDEX/FALL '82が開催された。COMDEX(COMputer Dealer's EXposition)はアメリカにおけるコンピュータの展示会の一つで，日本のマイコンショーにあたる。ただし規模はマイコンショーの数倍にもなり，アメリカコンピュータ産業の底力を感じさせる。

今回の出展企業はハード，ソフト，関連会社含め約1000社におよんだ。日本のショーはハード先行型で，どちらかというとハード屋が頑張っている印象だが，アメリカの場合はソフトのブースが大きい。これはソフト尊重の傾向をよく表しており，ソフトの本当の価値を理解しているものと思われる。

これからのソフトの動向としていくつか感じた点を述べる。ビジネスソフトは，データベース・スプレッドシート・ワードプロセッサが中心となり，それぞれのソフトを組み合せたり，ユニットとして使えるものが増え，ゲームソフトでは，動きの激しいもの・シミュレーションゲーム，が増えるのではないだろうか。

ハードでは，ポータブルコンピュータ・取りはずし可能の5MBのハードディスク，が注目株だ。

さて富士通のブースにはFM-16(micro16s)が4台，FM-7が2台出品されていた。FM-16は日本では未発表のものだが，FM-11とデザインがよく似ている。初めはただのマイナーチェンジか，と思ったが……。

ハードウェアをみると……

メインCPUは 8086とZ80をデュアルで採用し，CP/M-86™，MS-DOS™，CP/M®-80，Concurrent CP/M™すべてがオプションなしで走る。サブCPUに6809が入っているが，メインに入っていないのは6809ファンとしてはちょっと寂しい。しかし，68K，80186，Z-8000のCPUボードがサポートされる予定だ。

メモリは標準128Kバイトで，パリティチェック可能である。

I/O関係は……

目新しいものとして 10MBのウインチェスタハードディスクを電源の増設なしで，即セット可能である。また，オプションでCorvas社のOmninet™を採用している。

ソフトウェア……

ワードプロセッサ，スプレッドシートプログラム(Super Calc™)が標準として用意されている。

全体的なイメージ……

本格的16ビットビジネスマシンで，オーソドックスな設計となっている。日本で市販されるのはまだ先のようだが，FMファミリーの最上位機種としてアメリカでの健闘を期待したい。

以下にスペックを紹介する。

FM-16のハードウェア

プロセッサ
- 8086, Z80A　デュアルメインプロセッサ
- 68B09E　ビデオ用サブプロセッサ
- MB88401　キーボード用CPU
- 68000　ソフトカード(オプション)
- 68B09E　ソフトカード(オプション)

メモリ

ユーザーRAM 128Kバイト(パリティ付き)，拡張時1Mバイト

ビデオRAM 52Kバイト

画面仕様
- キャラクタ2,000文字
キャラクタごとに8色
- 640×200ドットグラフィック
ドットごとに8色
- キャラクタ画面とグラフィック画面混在可

外部メモリ

ミニフロッピーディスク2台標準装置(320KB×2)

5¼インチウィンチェスタディスク(オプション)

キーボード

薄型スカルプチャタイプ

インタフェース(標準実装のもの)
- プリンタインタフェース(セントロニクス仕様)
- RS-232C(同期／非同期)
- カラーRGB, CRTインタフェース
- 白黒CRTインタフェース
- 4チャンネルA／Dコンバータ
- ライトペンインタフェース

ソフトウェア

システムソフト

標準実装　CP／M-86(8086)

オプション

CP／M3.0(Z80用)

Concurrent CP／M(8086用)

MS-DOS(8086用)

アプリケーションソフト

・スーパーカルク
・ワードプロセッサ
・CP／M-80, CP／M-86, MS-DOSで走る全ソフト

ペリフェラル（外部機器）

ミニフロッピーディスク2台：5¼インチ320KB／1台
ウィンチェスタドライブ：5¼インチ10MB
カラーCRT：12インチ640×200ドット8色
白黒CRT：12インチ640×200ドット
プリンタ：
　ドットマトリクス式　MX－80エプソン
　高品質プリンタ　ブラザー HR-1
ミニフロッピー2台とRGB CRTは標準装備

ハードウェアスペック

1．CPU
メイン　8086　8MHz
サブ　68B09E　2.0MHz
キーボード　MB88401　4.2MHz

2．メモリ
メイン部ユーザーRAM
　128Kバイト（パリティ付き）
　拡張時最大1Mバイト
サブ部
　グラフィックビデオRAM　48Kバイト
　キャラクタVRAM　4Kバイト
　モニタ用ROM　10Kバイト

3．ブートローダ，フロントパネルの機能
ブートローディングは，フロントパネルのスイッチにより選択可
　5¼インチフロッピー，8インチフロッピー，ハードディスク，ボーレート，CRT極性変更可

4．キーボード
・本体から分離　カールコードで接続
・薄型スカルプチャキーボード
・キートップはシリンドリカルスカルプチャ方式，98キータイプライタ準拠
・MB88401によるソフトウェアスキャン，Nキーロールオーバ，レポート機能付き
・CAP…キャピタルロック機能（LED付き）
・INS……インサートモード（LED付き）

5．CRTディスプレイ
・キャラクタ221種
・8×8ドットマトリクス
・ディスプレイモード　80×25, 40×25選択
・白・黒・赤・緑・黄・青・紫・シアンの8色をキャラクタ単位で指定可
・機能
　アップ，ダウン，左右スクロール
　カーソル機能（反転，点滅）
　スクリーンエディット機能
　コントロールコードキャラクタ可能
　グラフィックス640×200，8色／1ドット

6．ブザー
1kHz　ON／OFF

7．タイマ
20msタイマ（NMIインタラプトがサブCPUにかかる）

8．DMA機能
4チャンネルがフロッピーディスク用に使用されている

9．割り込み
メインCPU
・NMI…パリティチェック　DMAの時間オーバ
・FIRQ…拡張時使用可，ブレークキー，応答用
・IRQ…RS-232C，キー入力，拡張用，DMA，ミニFDD，プリンタ
サブCPU
・NMI…内部タイマ（20ms）
・FIRQ…ライトペン
・IRQ…メインからのキャンセルコール

10．インタフェース
・ミニフロッピードライブ
　DMAにより高速転送
・ライトペン
・オーディオカセットインタフェース
　1,200ボー（平均），リモート機能付き
・アナログ入力端子（4チャンネル）
　入力電圧　ハイ→0～2.5V
　　　　　　ロー→0～0.625V
　8ビットに変換，5ms
・RS-232C＝EIA　RS232C-24
　転送スピード　300/600/1200/2400/4800/9600ボー（フロントパネルスイッチで切り換え）
　IRQ割り込み　マスク設定可能
　同期，非同期
・プリンタ
　8ビットパラレル（TTLレベル）
　セントロニクス，アスキーコード
・拡張バス
　130ピン拡張ポート　5スロット
　ボードの大きさ：4¼"(H)×8"(D)×⅛"(W)

11．電源
入力110V　50／60Hz

12．設置条件
温度：0～35℃　湿度：20～80%

13．大きさ
本体：19¼"(W)×14½"(D)×5¾"(H)
キーボード：18"(W)×7½"(D)×1½"(H)

14．重さ
15kg（キーボード含む，FDD含まず）

CP／M®，CP／M-86™，Concurrent CP／M™はDigital Research社，MS-DOS™はMicrosoft社，Supercalc™はSorcim社，Omninet™はCorvas社の登録商標です。

新製品情報

エプソン／諏訪精工舎

FP-80，RP-80を新発売

エプソン㈱と㈱諏訪精工舎は，ターミナルプリンタMPシリーズの後継機種として，新たにFP-80シリーズ，RP-80シリーズを開発，販売を開始した。

FP-80シリーズは，160cpsとMPシリーズの2倍の印字速度をもち，9種類の超高精度ビットイメージプリントが可能，アンダーライン機能，スーパー／サブスクリプト文字，強調文字などの文字種に加え，国際文字印字（9か国対応），ダウンロード文字（ユーザー定義可能），プロポーショナル印字，エリート印字も可能，などの特徴をもち，価格は149,800円（標準機）とMPシリーズ並み。

RP-80シリーズは，100cpsの印字速度（MPシリーズの1.25倍），6種類の超高精度ビットイメージプリントが可能などの特徴のほか，従来のMPシリーズの機能をすべてカバーして，89,000円と超低価格を実現した。

投稿記事

タイプ練習プログラム

画面写真は7ページにあります。

宇久一輝

マイコンを使ってみて，キーボードの練習を，と考える人は多いと思う。タイプ練習プログラムは，今までにも多くあるが，どのプログラムも今ひとつ，ものたりないような気がする。単にA，S，D，F，A，S，D，F…と打っていけば覚えるというものではない。私自身，それではすぐあきてしまった。そこで，1日30分，10日練習することによって，キーボードを見ないで打てるようになるプログラムを私なりに組んでみた。

A～Zの英大文字と，@，／，カンマ，コロン，セミコロン，ピリオドなどを10ブロックに分け，1単元30分，10日で10単元終了とした。また，1日の練習の最後に結果をカセットにセーブし，翌日はそれをロードしてその単元から練習を続けられるようにした。

使用法

このプログラムを動かすには，本体(FM－8または7)，カセットレコーダ，CRTのほかに，漢字ROMが必要となる。漢字ROMを持っていない人は，リスト2のように各行を変更する。また，ディスクベースの場合は，文番号1310のREMをはずし，KILL　命令を働かせること。

まずプログラムを入力後，RUNさせると，初めてかどうか聞いてくる。Y⤵(Returnキー)と打ち込むと，デモ表示が出るので，心構えができたらスペースバーを押す。もう一度キーボードが表示され，今回練習するキーは紫色で表示される。Returnキーを押すと練習開始となり，画面上方の四角いワクの中に押すべき文字が表示されるので，その文字を入力する。英小文字で入力しても受け付けないので，CAPキーはロックすること。間違わずに時間内に入力できれば，ワクの中の文字は緑地に赤で表示され，ビープ音がでる。

練習終了とともにメッセージが表示され，練習を続けるかどうか聞いてくるので，Y⤵orN⤵で答える。もしこのレベルに合格したら，次の単元に進むことができる。N⤵で練習を打ち切ると，今回のデータをカセットにセーブして，メッセージとともにプログラムを終了する。

2日目からは，初めてかどうか聞いてきたときにN⤵または単に⤵で答える。前回のデータをロードして，その単元から練習を続ける。

全ての単元（10単元）を終了すると，メッセージを出力してプログラムを終了する。

このように，いちいち練習の結果をセーブするため，ロスの少ないレッスンができる。また，CRTにキーボードが表示されるので，目を下にもっていかないクセがつき，キーボードを見ないで打てるようになるはずである。

ユーザー変更箇所

このプログラムは，次の各行を変更することにより，自分に合った練習プログラムに直せる。

●1170　正解率

行番号1170で，次の単元に進める正解率を規定している。現在は正解率80％を超えると進級できるが，完璧を求める人は「.999999」

にして100％正確に入力しないかぎり，進級できないようにもできる。

●1180　練習量

1つのキーに対する練習量をおおよそ10回としたが，ものたりない人は，もっと増やしてもよい。

●1190　練習開始のレベル（1～10）

レッスン終了後，セーブを忘れたときなどは，行番号1190の1という数字を直してからRUNさせればよい。

●1800　制限時間

行番号1800の44という数字を変えることでキーを受け付ける時間が変更できる。25から60ぐらいがてごろだが，超初心者は100ぐらいでもよいだろう。

●2290～2330　キーの順番

行番号2290～2330のDATA文中のダブルコーテーションで囲まれた文字を変更することにより，練習するキーの順番が変えられる。

●2350　1単元で増やすキーの量

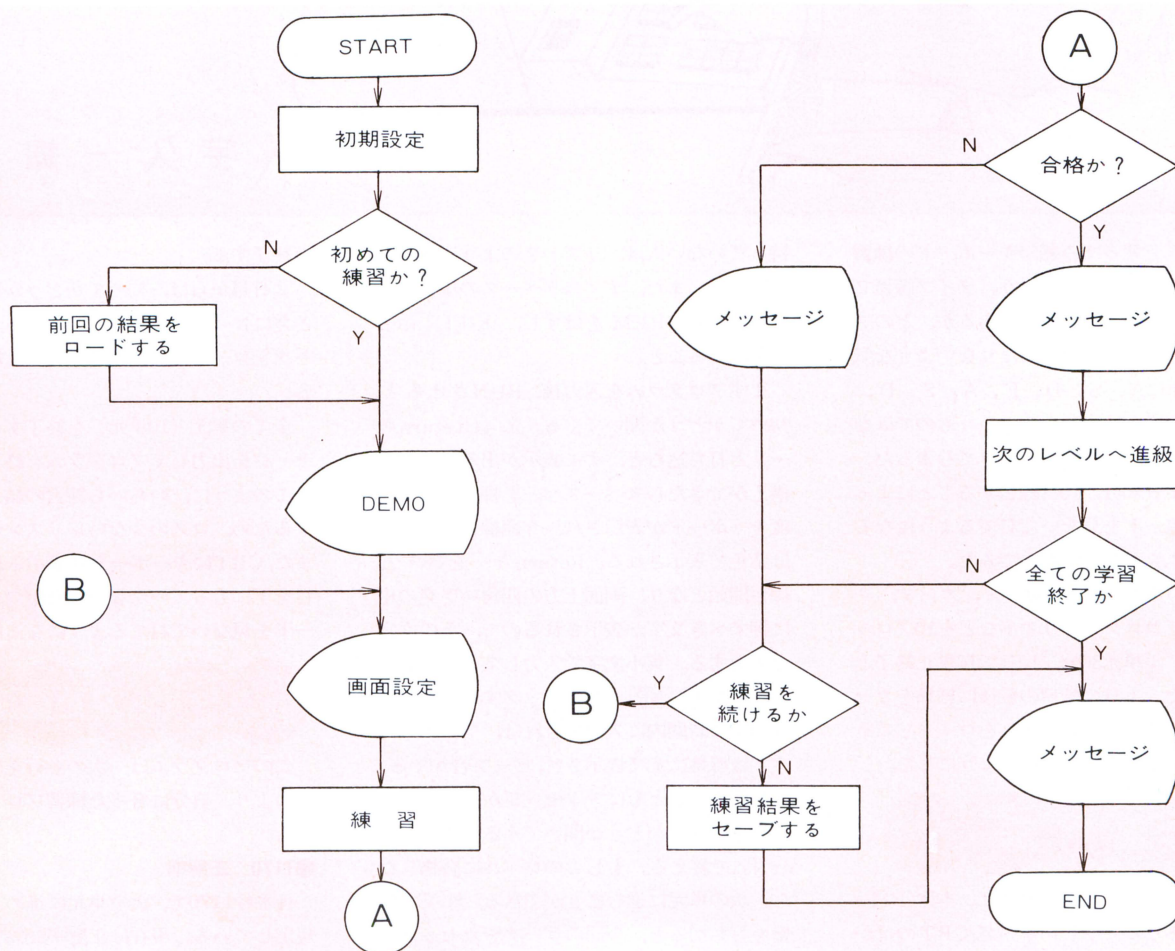
最初の第1単元では4個のキーの練習，第2単元ではさらに4個のキーを追加して，8個のキーを練習，というようにしてある。この行番号2350のデータを変えることによって，各単元で追加するキーの数を変えられる。

▼図2　キーボードのASDFとJKL；はホームポジション

*　*　*

以上の変更で，マイペースのレッスンが可能になるので，タイピングにかかるむだな時

▼図1　フローチャート

START
初期設定
初めての練習か？
N
前回の結果をロードする
Y
DEMO
B
画面設定
練　習
A
A
合格か？
N
Y
メッセージ
メッセージ
次のレベルへ進級
全ての学習終了か
N
Y
B
Y
練習を続けるか
N
練習結果をセーブする
メッセージ
END

間を節約して，一日もはやくFMを使いこなそう。

キーボードのデモ表示もきれいだし，打ち込んだキーが正しいかどうかも，色や音で教えてくれてわかりやすくできている。特に練習の結果，どの単元まで進んだかカセットにいれるアイデアもいい。

ただ，タイピングはいい加減な指で打っていると，いつまでたっても上達しない。正しい指遣いをマスターするためには，行番号2290〜2330は，F，G，H，J……ではなく，D，F，J，K，A，S，L，；……とでもした方がよいだろう。なお，ASDFとJKL;の所はホームポジションと言って打ち終えた後，必ずこの位置に指を置いおく（図2）。

▼リスト１（漢字ROM使用）


```
1000 ' ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
1010 ' ■                                                                       ■
1020 ' ■                                                                       ■
1030 ' ■        KEY BOARD LESSON PROGRAM  v1.0                  S.1981          ■
1040 ' ■                                                                       ■
1050 ' ■━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━■
1060 '
1070 '■  INITIALIZE ROUTINE
1080 WIDTH 80,25:CLEAR 500,&H4FFF
1090 PRINT@ (188,92),&H3E2F,0,&H2139,0,&H242A,0,&H4254,0,&H2441,0,&H323C,0,
         &H2435,0,&H2424,0,&H2123
1100 DIM CM(47,2)        ' CARACTER MAP &H2C-&H5A
1110 DIM LV(10)          ' ﾚﾍﾞﾙ        1-10
1120 DIM KM(32)          ' ｷｰ ﾉ ｱﾂｶｳ ﾚﾍﾞﾙｻ
1130 FOR L=1 TO 47:READ WK:CM(L,1)=INT(WK):CM(L,2)=(WK-INT(WK))*30:NEXT
1140 FOR L=1 TO 32:READ WK$:KM(L)=ASC(WK$)-&H2B:NEXT
1150 SUM=0:FOR L=1 TO 10:READ WK:SUM=SUM+WK:LV(L)=SUM:NEXT
1160 '
1170 RITS = .8 ' ｼﾝｷｭｳ ｽﾙ ﾀﾒﾉ ｾｲﾄｳ ﾘﾂ ﾊ 80 % ﾃﾞｽ.
1180 KAISU= 10 ' 1 ﾂﾉ ｷｰ ﾆ ﾀｲｽﾙ ｵｵﾖｿ ﾉ ﾚﾝｼｭｳ ｶｲｽｳ. ( 10 ｶｲ )
1190 V2   =  1 ' ﾊｼﾞﾒﾉ ﾚﾍﾞﾙ
1200 '■  DATA LOAD
1210 CLS
1220 LOCATE 10,12:LINE INPUT "IS THIS PROGRAM FIRST [Y/N RETURN] ";WK$
1230 LOCATE 10,12:PRINT STRING$(40," ");
1240 IF WK$="Y" OR WK$="y" THEN 1340
1250    PRINT@(48,80),&H252B,0,&H253B,0,&H2543,0,&H2548,0,&H2546,0,&H213C,0,
          &H2557,0,0,&H244E,0,&H3D60,0,&H4877,0,&H242C,0,&H3D50,0,&H4D68,0,
          &H245E,0,&H2437,0,&H243F,0,&H2469
1260    PRINT@(48,100),&H2352,&H2345,&H2354,&H2355,&H2352,&H234E,0,&H252D,0,
          &H213C,0,&H2472,0,&H3221,0,&H2437,0,&H2446,0,&H323C,0,&H2435,0,&H2424
,         0,&H2123
1270    LOCATE 78,23:LINE INPUT WK$
1280    PRINT@(160,120),&H234C,0,&H234F,0,&H2341,0,&H2344,0,0,&H4366
1290 OPEN "I",1,"KDATA":INPUT#1,V2:CLOSE #1
1300 '
1310   REM KILL"KDATA"          ' ﾃﾞｨｽｸ ﾊﾞﾝ ﾉ ﾄｷ REM ｦ ｻｸｼﾞｮ
1320 '
1330 '■  DEMO
1340 CLS
1350   FOR L=1 TO 32
1360     WK=KM(L):X=CM(WK,1):Y=CM(WK,2):IF X+Y=0 THEN 1390
1370       LOCATE X,Y:PRINT CHR$(WK+&H2B);
1380       X=X*8:Y=Y*8:LINE(X-10,Y-4)-(X+18,Y+12),PSET,7,B
1390   NEXT
1400   PRINT@(37,0),&H4E7D,0,&H3D2C,0,&H332B,0,&H3B4F,0,&H244A,0,&H2469,0,0,
         &H2539,&H255A,&H213C,&H2539,&H2550,&H213C,0,0,&H2472,0,&H3221,0,&H2437
,        0,&H2446,0,&H323C,0,&H2435,0,&H2424,0,&H2123
1410   L=INT(RND*32+1)
1420   WK=KM(L):X=CM(WK,1):Y=CM(WK,2)
1430   X=X*8:Y=Y*8:LINE(X-9,Y-3)-(X+17,Y+11),XOR,7,BF
1440 IF INKEY$=" " THEN 1470 ELSE 1410
1450 '
1460 '■  SCREEN SET
1470 LOCATE 0,24:FOR L=1 TO 25:PRINT:NEXT
1480 PRINT@(152,0),&H3C21,0,&H244E,0,&H3D67,0,&H4856,0,&H2447,0,&H4E7D,0,&H3D2C
```

```
,       0,&H2437,0,&H245E,0,&H2439,0,&H2123
1490   FOR LU=1 TO 10:FOR L=1+LV(LU-1) TO LV(LU)
1500       WK=KM(L):X=CM(WK,1):Y=CM(WK,2):IF X+Y=0 THEN 1550
1510       LOCATE X,Y:PRINT CHR$(WK+&H2B);
1520       X=X*8:Y=Y*8:LINE(X-10,Y-4)-(X+18,Y+12),PSET,7,B
1530       IF LU<=V2 THEN LINE(X-10,Y-4)-(X+18,Y+12),XOR,3,BF
1540       BEEP1:FOR M=0 TO 20:NEXT:BEEP0:FOR M=0 TO 100:NEXT
1550     NEXT
1560   NEXT
1570   LINE(0,0)-(639,15),PRESET,,BF
1580   PRINT@(120,180),&H3E65,0,&H244E,0,&H4A38,0,&H3B7A,0,&H2472,0,&H253F,&H2524
,        &H2557,0,&H2437,0,&H2446,0,&H323C,0,&H2435,0,&H2424,0,&H2123
1590   LINE (288,0)-(352,32),PSET,7,B
1600   LINE (290,1)-(350,31),PSET,6,B
1610   LINE (292,2)-(348,30),PSET,5,B
1620   LINE (294,3)-(346,29),PSET,4,B
1630   LINE (296,4)-(344,28),PSET,3,B
1640   LINE (298,5)-(342,27),PSET,2,B
1650   LINE (300,6)-(340,26),PSET,1,B
1660   LOCATE 0,0:LINE INPUT "Hit Key to start "; GK$
1670   LOCATE 0,0:PRINT STRING$(25," ");
1680  '■  LESSON
1690   SUCCESS=0
1700   N=0
1710     LOCATE 0,0:PRINT "ｱﾄ ";(11-V2)*30;"分 ｶﾞﾝﾊﾞﾛｳ!";
1720     LOCATE 40,20:PRINT "LEVEL  [";V2;" ]";
1730   WHILE N<V2*KAISU
1740     N=N+1
1750     MIN=1:MAX=LV(V2)
1760     RM=KM(   INT( RND*MAX+1 )   )
1770     FOR L=1 TO 1000:NEXT:PRINT CHR$(7);
1780     SYMBOL (304,9),CHR$(RM+&H2B),4,2,6
1790     G$=INKEY$:GK$=""
1800     F=0:FOR L=1 TO 44:G$=INKEY$:
         IF F=0 THEN IF G$<>"" THEN F=1:L=50:GK$=G$ ELSE NEXT ELSE NEXT
1810     G$=INKEY$
1820     IF GK$=CHR$(RM+&H2B) THEN SUCCESS=SUCCESS+1:GOSUB 2080
1830     LINE(304,8)-(336,24),PRESET,,BF
1840     LOCATE 0,20:PRINT "SCORE  [";SUCCESS*10;" ]";
1850   WEND
1860  '■  NO SUCCESS
1870   IF SUCCESS>N*RITS THEN 1950
1880   CLS
1890     PRINT@(144, 50) ,&H243F,&H2440,&H2424,&H245E,&H244E,&H4E7D,&H3D2C,
           &H244E,&H376B,&H324C,&H244B,&H2124,&H423F,&H3E2F,&H4954,&H4B7E,&H242C,
           &H2422,&H246A,&H245E,&H2439,&H2123
1900     PRINT@(144, 70) ,&H2462,&H2426,&H3E2F,&H2437,&H242C,&H2473,&H2450,
           &H2443,&H2446,&H4E7D,&H3D2C,&H2437,&H2446,&H323C,&H2435,&H2424,&H2123
1910   FOR L=1 TO 4000:NEXT
1920   LOCATE 0,22:LINE INPUT "ﾚﾝｼｭｳ ｦ ﾂﾂﾞｹﾃ ﾔﾘﾏｽｶ?  [Y/N]";G$
1930   IF G$="Y" OR G$="y" THEN 1470 ELSE 2000
1940  '■  SUCCESS
1950   CLS
1960   PRINT@(112, 50) ,&H242A,&H2461,&H2447,&H2448,&H2426,&H2120,&H2434,&H2436,
         &H2424,&H245E,&H2439,&H2123
1970   PRINT@(112, 70) ,&H4E7D,&H3D2C,&H244E,&H376B,&H324C,&H242C,&H2439,&H2450,
         &H2469,&H2437,&H242B,&H2443,&H243F,&H244E,&H2447,&H2124,&H3C21,&H244E,
         &H4331,&H3835,&H2458,&H3F4A,&H2461,&H245E,&H2439,&H2123
1980   V2=V2+1:IF V2=11 THEN 2130 ELSE 1910
1990  '■  END.
2000   CLS
2010     PRINT@(48,80),&H252B,0,&H253B,0,&H2543,0,&H2548,0,&H2546,0,&H213C,0,
           &H2557,0,0,&H244E,0,&H3D60,0,&H4877,0,&H242C,0,&H3D50,0,&H4D68,0,
           &H245E,0,&H2437,0,&H243F,0,&H2469
2020     PRINT@(48,100),&H2352,&H2345,&H2354,&H2355,&H2352,&H234E,0,&H252D,0,
           &H213C,0,&H2472,0,&H3221,0,&H2437,0,&H2446,0,&H323C,0,&H2435,0,&H2424,
           0,&H2123
2030     LOCATE 78,23:LINE INPUT WK$
```

```
2040     PRINT@(160,120),&H2353,0,&H2341,0,&H2356,0,&H2345,0,0,&H4366
2050   OPEN "O",1,"KDATA":PRINT #1,V2:CLOSE 1
2060   GOTO 2130
2070 '■  SUB PROGRAM
2080   LINE (301,7)-(339,25),XOR,4,BF
2090   FOR L=1 TO 30:BEEP1:FOR M=1 TO 20:NEXT:BEEP0:NEXT
2100   LINE (301,7)-(339,25),XOR,4,BF
2110   RETURN
2120 '■  MES
2130   CLS
2140   PRINT@(112, 20) ,&H234C,&H2345,&H2353,&H2353,&H234F,&H234E,&H2120,&H244F,
         &H2120,&H2462,&H2426,&H2120,&H242A,&H246F,&H246A,&H2120,&H2447,&H2439,
         &H242B,&H2129,&H2120
2150   PRINT@(112, 40) ,&H2447,&H244F,&H2124,&H4959,&H3B4E,&H444C,&H2120,&H234D,
         &H2369,&H2363,&H2372,&H236F,&H2338,&H2120,&H244E,&H4545,&H383B,&H2472,
         &H405A,&H2443,&H2446,&H323C,&H2435,&H2424,&H2123
2160   PRINT@(112, 90) ,&H2354,&H2368,&H2369,&H2373,&H2120,&H2370,&H2372,&H236F,
         &H2367,&H2372,&H2361,&H236D,&H2120,&H236D,&H2361,&H2364,&H2365,&H2120,
         &H234B,&H2125,&H2355,&H234B,&H2355,&H2125
2170   END
2180   ::::::::::::::::::::::::::::::::::::::::::::' CM(x,y) <-- x,y
2190     DATA 58.5 , 0   ,65.5 ,72.5 , 0   , 0
2200     DATA  0   , 0   , 0   , 0   , 0   , 0
2210     DATA  0   , 0   ,77.4 ,70.4 , 0   , 0
2220     DATA  0   , 0   ,75.3
2230     DATA  7.4 ,37.5 ,23.5 ,21.4 ,19.3 ,28.4
2240     DATA 35.4 ,42.4 ,54.3 ,49.4 ,56.4 ,63.4
2250     DATA 51.5 ,44.5 ,61.3 ,68.3 , 5.3 ,26.3
2260     DATA 14.4 ,33.3 ,47.3 ,30.5 ,12.3 ,16.5
2270     DATA 40.3 , 9.5
2280   ::::::::::::::::::::::::::::::::::::::::::::' KM(x)    <-- ASC(x)-$2B
2290     DATA "F","G","H","J","R","T","Y","U"
2300     DATA "V","B","N","M","D","K"
2310     DATA "E","C","I",",","S","L","A",";"
2320     DATA "W","X","O",".","Q","Z","P","/"
2330     DATA "@",":"
2340   ::::::::::::::::::::::::::::::::::::::::::::' LV(x)    <-- SUM=SUM+x
2350     DATA 4,4,4,2,4,4,2,2,4,2
2360   ::::::::::::::::::::::::::::::::::::::::::::
```

▼リスト２（漢字ROMがない場合の変更箇所）

```
1090  SYMBOL(188,92),"ﾁｮｯﾄ ﾏｯﾃﾃ ｸﾀﾞｻｲﾈ.",2,2
1250      SYMBOL(96,80),"ｶｾｯﾄ ﾃｰﾌﾟ ﾉ ｼﾞｭﾝﾋﾞ ｶﾞ ﾃﾞｷﾏｼﾀﾗ",2,2
1260      SYMBOL(96,100),"'RETURN' ｷｰ ｦ ｵｼﾃ ｸﾀﾞｻｲ.",2,2
1280      SYMBOL(220,120),"LOAD ﾁｭｳ ﾃﾞｽ.",2,2
1400    SYMBOL(48,8),"ﾚﾝｼｭｳ ｶｲｼ ﾅﾗ ｽﾍﾟｰｽ ｷｰ ｦ ｵｼﾃ ｸﾀﾞｻｲ.",2,2
1480  SYMBOL(96,2),"ﾂｷﾞﾉ ｼﾞｭﾝﾊﾞﾝ ﾃﾞ ﾚﾝｼｭｳ ｦ ｼﾏｽ.",2,2
1580  SYMBOL(112,180),"ﾜｸ ﾉ ﾅｶ ﾉ ﾓｼﾞ ｦ ｳｯﾃ ｸﾀﾞｻｲ.",2,2
1890    SYMBOL(120,80) ,"ﾓｳ ﾁｮｯﾄ ｶﾞﾝﾊﾞｯﾃ ﾐﾃ ｸﾀﾞｻｲ.",2,2
1900    SYMBOL(120,100) ,"ｷｯﾄ ﾖｲ ｹｯｶ ﾆ ﾅﾙﾃﾞｼｮｳ.",2,2
1960  SYMBOL(140, 50) ,"ｵﾒﾃﾞﾄｳ ｺﾞｻﾞｲﾏｽ.",2,2
1970  SYMBOL(140, 70) ,"ﾚﾝｼｭｳ ﾉ ｹｯｶ ｶﾞ ﾖｶｯﾀ ﾉﾃﾞ",2,2
1975  SYMBOL(140, 90) ,"ﾂｷﾞ ﾉ ﾀﾝｹﾞﾝ ﾍ ｽｽﾒﾏｽ.",2,2
2010      SYMBOL(96,80),"ｶｾｯﾄ ﾃｰﾌﾟ ﾉ ｼﾞｭﾝﾋﾞ ｶﾞ ﾃﾞｷﾏｼﾀﾗ",2,2
2020      SYMBOL(96,100),"'RETURN' ｷｰ ｦ ｵｼﾃ ｸﾀﾞｻｲ.",2,2
2030      LOCATE 78,23:LINE INPUT WK$
2040      SYMBOL(220,120),"SAVE ﾁｭｳ ﾃﾞｽ.",2,2
2140    SYMBOL(112,60) ,"LESSON ﾊ ﾓｳ ｵﾜﾘ ﾃﾞｽｶ?",2,2
2150    SYMBOL(112,80) ,"ﾃﾞﾊ FM-8 ﾉ ﾃﾞﾝｹﾞﾝ ｦ ｷｯﾃ ｸﾀﾞｻｲ.",2,2
2160 （削除）
```

パソコンでテレビスタジオと家庭を直結
“ザ・コンピニオン”

KEY LAB 中之島
赤松　則

写真1　データテレホン
右はしの溝は，磁気カード
を通すためのもの

複数の電子計算機を結ぶコンピュータネットワークが注目されている。このシステムをテレビ番組に利用しているのが，“ザ・コンピニオン”（朝日放送，全国20局ネット）。家庭とスタジオをコンピュータで結び，視聴者の考え方が，すぐに番組に反映される新しいタイプの視聴者参加番組だ。この全国規模の“双方向テレビ番組”も，放送開始から約1年。コンピュータを“大道具”として定着させた。

システム

テレビは，放送局から各家庭への“一方通行”の情報伝達だけに，これまで，視聴者の意見を番組に反映させるために，いろいろな方法がとられていた。ハガキや電話の利用，視聴者にスタジオに集まってもらう方法などである。しかし，どれも，全国規模で，数百人のデータをリアルタイムで集計するには，かなりの無理がある。

そこで登場したのが，コンピュータのファイルを投票箱の代わりに使おうというアイデアだ。この方法を利用すれば，日本各地に分散する数百人のデータを数分で集計し，スタジオでグラフ化することができる。

しかし，実現には，いくつかの問題もあった。

まず，投票箱とするコンピュータである。これには、数十人の人が同時に接続し，データを入力することになる。また，各家庭におおげさなコンピュータターミナルを設置するわけにもいかない。さらに，蓄積，集計されたデータをそのままスタジオに転送し，瞬時にグラフ化することなども必要になる。

そのうち，データの入力と集計は，大型計算機のTSS（Time Sharing System）の一種である電電公社のDRESS（Denden kosha REaltime Sales management System：電電販売在庫管理システム）と，プッシュホンにコンピューターターミナルとしての機能を付けたデータテレホンを利用することにより解決した（写真1）。

また，大型計算機からデータを取り出したり，それをグラフ化するためにスタジオ内に設置するコンピュータは，優れた通信回線制御機能とグラフィック機能を持つ上，移動も簡単な，富士通のパーソナルコンピュータが選ばれた。

このようにして，でき上がったのが，コンピニオンのシステムである（図1）。

スタジオにずらりと並んだ十数台のFMシリーズは，全国五箇所のDRESSセンタ，さらに，それにつながる300世帯のデータテレホンと電話回線で結ばれている（写真2）。電話からの音声信号を電気信号に，そして逆に，パソコンからの電気信号を音声信号に変えるための音響カプラがその接点といえるであろう（写真3）。

日本全国11ブロック（札幌，仙台，福島，東京，静岡，名古屋，大阪，岡山，広島，高松，福岡）から選ばれた300世帯のモニタ家庭には，1世帯平均4人の家族があり，その一人一人は固有の視聴者番号を記憶させた番組参加用の磁気カード（写真4）を持っているので，約1200人からの意見を集めることができる。

スタジオ内のFM-8は，RS-232Cインタフェースを利用して結ばれ，画面は本体付属のカラーCRTのほか，スタジオパネルの26インチカラーモニタや副調整室のモニタにも写し出される。また，その一部は，そのまま電波に乗って各家庭のブラウン管に表示される。

ちょっと見ただけでは，いかにも難しそうなシステムに感じられるが，そのほとんどは自動化されているので，操作はいたって簡単である。キーボードに触るのは初めてという女性でも，仕事の流れさえわかれば，1週間もあれば，充分にマスターできる。ただ，生放送だけに失敗は許されない。

写真2　ずらりと並んだFMシリーズ
コンピニオンのデータを瞬時
に処理している

番組とコンピュータ操作

では，実際の番組で，このシステムがどのように使われているのかを紹介しよう。

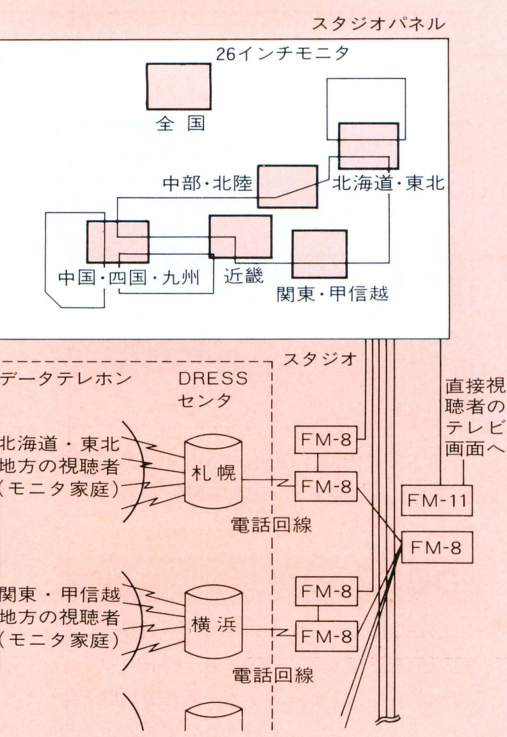
パソコンの操作には6人のコンピニオンガールがあたり，一人で，ファイル転送用と画面表示用の2台を受け持つ。

日曜の朝8時過ぎ，彼女たちの仕事はまず，パソコン間の接続確認。つづいて，電話器でDRESSセンタを呼び出し，受話器を音響カプラにセットした後，DRESSにパスワード（暗照番号）を送れば，すべての接続は完了する。

これだけの操作で，コンピニオンのコンピュータシステムは動き始め，あとは，本番を待つばかりである。

8時半，軽快なテーマソングとともに番組開始。ブラウン管を通じて，楽しい，また時にはシビアな質問が視聴者に問いかけられる。全国300のモニタ家庭の人たちは，これに対して，自分の考えを質問ごとに指定された回答番号（例えば，①すき②きらい③わからない）でDRESSに送る。

DRESSへの入力方法は，銀行でキャッシュカードを利用するのに似ている。まず，データテレホンの［準備］ボタンを押し，磁気カードを通す。このカードには，投票資格のある視聴者であることを示す暗号や，投票処理プログラムなどが入っているので，これで投票準備は完了。回答番号を押して，［要求］ボタンを押せば，データは自動的に大型計算機のファイルに蓄積される。もし，その質問に対して意見のある時は，さらに，［A］のボタンを押す。

図1　システム構成

ザ・コンピニオン

毎週日曜日　午前8時30分～9時

朝日放送(ABC)	名古屋放送(NBN)
北海道テレビ(HTB)	日本海テレビジョン放送(NKT)
青森放送(RAB)	広島ホームテレビ(UHT)
東日本放送(KHB)	山口放送(KRY)
秋田テレビ(AKT)	瀬戸内海放送(KSB)
福島放送(KFB)	愛媛放送(EDC)
テレビ朝日(ANB)	九州朝日放送(KBC)
テレビ信州(TSB)	テレビ熊本(TKU)
静岡県民放送(SKT)	鹿児島放送(KKB)
富山テレビ放送(T34)	沖縄テレビ放送(OTV)

ANN系列
ザ・コンピニオン
参加カード
放送日/毎日曜日AM8:30−9:00

写真4　番組参加用の磁気カード

出題後，投票は1分間で締め切られ，その結果は，各地区ごとにDRESSで集計される。

集計されたデータをスタジオに引き出してくるのは，各地区のDRESSを担当するFM-8である。コンピニオンガールの指示で，DRESSから転送されたデータは，すぐに表示用のFM-8に送られ，棒グラフや円グラフなどで，スタジオのパネルにセットされた26インチのカラーモニタに写し出される。

全国集計用のFM-8は，各地区を担当するFM-8からいっせいに送られてくるデータを，自動的に先着順に受け取り，外部記憶装置に蓄積する。

このデータは，全体集計のほか，男女別，年齢別，職業別などで，集計，グラフ化されスタジオモニタに表示されるとともに，直接，電波に乗って各家庭へ届けられる。

問いかけからグラフ表示までの，一連の処理に要する時間は，ほんの2，3分。質問について，スタジオや家庭で話し合っているうちに，結果はブラウン管に表示されるのである。

写真3　受話器をセットしてあるのが音響カプラ

投票の時に，意見アリのボタン［A］を押した視聴者の名前や電話番号は，各地区担当のFM-8，さらに，ディレクタのところに置いてあるモニタで知ることができ，電話で生の声を聴くことも可能である。

〝ザ・コンピニオン〟は，アップトゥデートな話題をブラウン管を通じて，全国の視聴者に問いかけ，全国のモニタ家庭からの声は，瞬時にスタジオに集められ，グラフ化，そして再び電波に乗って家庭へ，という大型計算機のTSSとパーソナルコンピュータの通信回線制御機能，グラフィック機能を生かした新しいタイプのテレビ番組だ。

さらにこの番組では，コンピニオンが始まるまではパソコンに触れたこともなかった，コンピューター一年生ともいえるスタッフが，毎週，パソコンやデジタイザを駆使して，漢字によるグラフの標題や質問用の画面，簡単なイラストを作成している。

簡便である上，汎用機としてのすぐれた機能と多くの可能性を持つパソコンは，今後，放送の世界でもさらに幅広く応用されていきそうだ。

パソコンと大型計算機および異機種パソコンとの通信は，このように日常的なものである。KEY LABでは，このためのTSSプログラムおよびファイル交換プログラム（FTR）を発売している。

価格は，どちらも￥12,000（各機種，ディスク，テープの両タイプとも）。

■お問い合わせは
〒530　大阪市北区中之島4－3－36
KEY　LAB　中之島まで

FLEX® 入門

西村義孝

図 2

```
$CD33 (SETEXT)     SET EXTENSI
$CD00 (COLDS)      COLDSTART ENTRY POINT
$CD03 (WARMS)      WARMSTART ENTRY POINT
$CD06 (RENTER)     DOS MAIN LOOP RE-ENTRY POINT
$CD09 (INCH)       INPUT CHARACTER
$CD0C (INCH2)      INPUT CHARACTER
$CD0F (OUTCH)      OUTPUT CHARACTER
$CD12 (OUTCH2)     OUTPUT CHARACTER
$CD15 (GETCHR)     GET CHARACTER
$CD18 (PUTCHR)     PUT CHARACTER
$CD1B (INBUFF)     INPUT INTO LINE BUFFER
$CD1E (PSTRNG)     PRINT STRING
$CD21 (CLASS)      CLASSIFY CHARACTER
$CD24 (PCRLF)      PRINT CARRIAGE RETURN AND LINE FEED
$CD27 (NXTCH)      GET NEXT BUFFER CHARACTER
$CD2A (RSTRIO)     RESTORE I/O VECTORS
$CD2D (GETFIL)     GET FILE SPECIFICATION
$CD30 (LOAD)       FILE LOADER
$CD33 (SETEXT)     SET EXTENSION
$CD36 (ADDBX)      ADD B-REGISTER TO X-REGISTER
$CD39 (OUTDEC)     OUTPUT DECIMAL NUMBER
$CD3C (OUTHEX)     OUTPUT HEXADECIMAL NUMBER
$CD3F (RPTERR)     REPORT ERROR
$CD42 (GETHEX)     GET HEXADECIMAL NUMBER
$CD45 (OUTADR)     OUTPUT HEXADECIMAL ADDRESS
$CD48 (INDEC)      INPUT DECIMAL NUMBER
$CD4B (DOCMND)     CALL DOS AS A SUBROUTINE
$CD4E (STAT)       CHECK TERMINAL INPUT STATUS
```

図 3


```
                 CD03  warms   equ    $cd03
                 CD18  putchr  equ    $cd18
  C100                         org    $c100
                 C100  main    equ    *
  C100 86    20                lda    #$20
                 C102  loop    equ    *
  C102 BD    CD18              jsr    putchr
  C105 4C                      inca
  C106 26    FA                bne    loop
  C108 7E    CD03              jmp    warms
                               end    main

 0 ERROR(S) DETECTED

 SYMBOL TABLE:

 loop    C102   main    C100   putchr  CD18   warms   CD03
```

FLEX®は，アメリカのテクニカル・システムズ・コンサルタンツ社で開発された68系の，シングルユーザー・シングルタスクのOS（オペレーティングシステム）である。マイコンのOSとしては80系のCP/M®が有名だが，68系のCP/Mと呼ばれているのが，このFLEXである。前回紹介されたOS-9™に比べると，FLEXは，OSとしては見おとりする部分もあるが，ソフトの蓄積という面から見ると，FLEXのほかには68系OSは考えられない。FLEXは，シングルユーザー・シングルタスクのOSだが，マルチユーザー・マルチタスクのFLEXであるUni FLEX®も発表されている。FLEXは，68系のOSだが，FLEXを使っている人の大半は6809ユーザーである。FLEXの動くハードウェアは，CP/Mに比べると多いとはいえないが，異機種間の互換性や，エディタアセンブラ，デバッガがそろっていることを考えると，FLEXは，かなり魅力的なOSといえるだろう。

移植性がよいのもFLEXの特長といえる。最低の環境としては，

0000～ 3FFFH （ユーザーエリア）

C000～ DFFFH （OS用）

のRAMおよび，2台のフロッピーがあれば，FLEXは動く。そのため，自作ボードや，シングルボードの6809システムでFLEXを採用している場合が多い。

FLEXの動くパソコン（というよりも，6809をCPUに使ったパソコン）があまりないため，FLEXを具体的に使ったことのある方は少ないと思うので，今回はFLEXの概要を説明しよう。

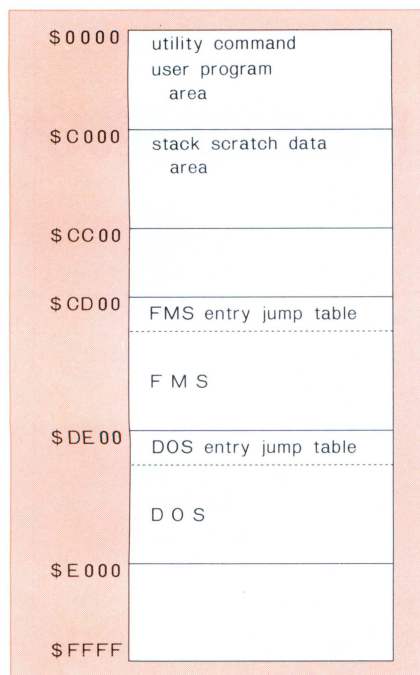
FLEXの構造

FLEXの内部はDOS，FMS，UCSの三つの部分に大別される（図1参照）。それらを紹介していこう。

DOS （ディスクオペレーティングシステム）

DOSは，ユーザーのコマンド列を解析し，処理する部分で，ユーザーと，FMS（ファイルマネージメントシステム…次項）の間とのコミュニケーションを行う部分である。エラ

図1　標準的なFLEXのメモリマップ

Address	Area
$0000	utility command user program area
$C000	stack scratch data area
$CC00	
$CD00	FMS entry jump table FMS
$DE00	DOS entry jump table DOS
$E000	
$FFFF	

一報告なども，DOSによって行われるので，一文字入出力，LINE入力などのルーチンが，ぎっしりと詰まっている。それらのエントリポイントを図2に示す。このように，FLEXシステムに共通な，エントリポイントを利用して，プログラムすることにより，異機種上で開発されたプログラムも実行できるわけだ。DOSルーチンを直接コールする簡単なデモプログラムを図3に示す。

FMS（ファイルマネージメントシステム）

FMS（ファイルマネージメントシステム）は，DOSとディスクとのコミュニケーションを行うもので，ファイルをオープンしたり，任意セクタにデータを書き込んだりといったディスク関係のことをすべて管理しているところである。DOSとFMSとの通信は，FCB（ファイルコントロールブロック）という部分を介して行われる。FCBは，F-BIOSにも同様のものがあるので，御存知の方も多いと思う。ユーザーは，FMSへのパラメータ列を，メモリ上のどこかに書く。そのパラメータ列が，FCBである。このままでは，FMSにはFCBがどこにあるかわからないので，FCBのトップアドレスをXレジスタにセットしてFMSをコールする。FCBは320バイトの長さがあり，そのフォーマットは図4のようになっている。byte0のFunction Codeは，FMSに対して「何をせよ」と命令するバイトで，Functionの一覧を図5に示す。FMSのエントリポイントは図6のようになっている。

図4

```
byte0        Function code
byte1        Error Status Byte
byte2        Activity Status
byte3        Drive Number
byte4-11     File Name
byte15       File Attributes
byte16       Reserved for future system use
byte17-18    Starting disk address of the file
byte19-20    Ending disk address of the file
byte21-22    File Size
byte23       File Sector Map Indicator
byte24       Reserved for future system use
byte25-27    File Creation Date
byte28-29    FCB list Pointer
byte30-31    Current Position
byte32-33    Current Record Number
byte34       Data Index
byte35       Random Index
byte36-46    Name Work Buffer
byte47-49    Current Directory Address
byte50-52    First Deleted Directory Pointer
byte53-63    Scratch Bytes
byte59       Space Compressing Flag
byte64-319   Sector Buffer
```

図5

```
FUNCTION 0 READ/WRITE NEXT BYTE/CHARACTER
FUNCTION 1 OPEN FOR READ
FUNCTION 2 OPEN FOR WRITE
FUNCTION 3 OPEN FOR UPDATE
FUNCTION 4 CLOSE FILE
FUNCTION 5 REWIND FILE
FUNCTION 6 OPEN DIRECTORY
FUNCTION 7 GET INFORMATION RECORD
FUNCTION 8 PUT INFORMATION RECORD
FUNCTION 9 READ SINGLE SECTOR
FUNCTION 10 WRITE SINGLE SECTOR
FUNCTION 11 RESERVED
FUNCTION 12 DELETE FILE
FUNCTION 13 RENAME FILE
FUNCTION 14 RESERVED
FUNCTION 15 NEXT SEQUENTIAL SECTOR
FUNCTION 16 OPEN SYSTEM INFORMATION RECORD
FUNCTION 17 GET RANDOM BYTE FROM SECTOR
FUNCTION 18 PUT RANDOM BYTE IN SECTOR
FUNCTION 19 RESERVED
FUNCTION 20 FIND NEXT DRIVE
FUNCTION 21 POSITION TO RECORD N
FUNCTION 22 BACKUP ONE RECORD
```

図6

```
$D400        FMS INITIALIZATION
$D403        FMS CLOSE
$D406        FMS CALL

GLOBAL VARIABLES FOR FMS

$D409-$D40A        FCB BASE POINTER
$D40B-$D40C        CURRENT FCB ADDRESS
$D435              VERIFY FLAG
```

図7

FLEXの使い方

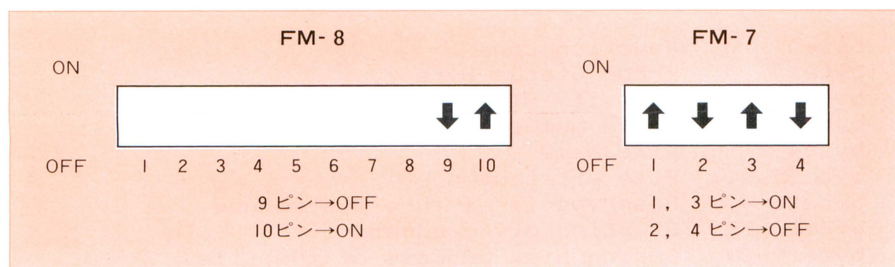



図8（FLEX FM-8用特記事項〈和文〉より引用）

1	FLEX	.COR		01-01	01-16	22	31-JUL-82
2	ERRORS	.SYS	R	01-17	01-1F	9	31-JUL-82
3	CAT	.CMD		01-20	02-02	3	31-JUL-82
4	COPY	.CMD		02-03	02-07	5	31-JUL-82
5	LIST	.CMD		02-08	02-0A	3	31-JUL-82
6	ASN	.CMD		02-0B	02-0B	1	31-JUL-82
7	DELETE	.CMD		02-0C	02-0D	2	31-JUL-82
8	RENAME	.CMD		02-0E	02-0E	1	31-JUL-82
9	TTYSET	.CMD		02-0F	02-10	2	31-JUL-82
10	P	.CMD		02-11	02-11	1	31-JUL-82
11	SAVE	.CMD		02-12	02-13	2	31-JUL-82
12	EDIT	.CMD		02-14	03-0F	28	31-JUL-82
13	ASMB	.CMD		03-10	04-1F	48	31-JUL-82
14	APPEND	.CMD		04-20	05-02	3	31-JUL-82
15	BUILD	.CMD		05-03	05-03	1	31-JUL-82
16	EXEC	.CMD		05-04	05-04	1	31-JUL-82
17	JUMP	.CMD		05-05	05-05	1	31-JUL-82
18	DATE	.CMD		05-06	05-07	2	31-JUL-82
19	O	.CMD		05-08	05-09	2	31-JUL-82
20	LINK	.CMD		05-0A	05-0A	1	31-JUL-82
21	VERSION	.CMD		05-0B	05-0B	1	31-JUL-82
22	PROT	.CMD		05-0C	05-0C	1	31-JUL-82
23	VERIFY	.CMD		05-0D	05-0D	1	31-JUL-82
24	PRINT	.CMD		05-0E	05-0F	2	31-JUL-82
25	QCHECK	.CMD		05-10	05-13	4	31-JUL-82
26	I	.CMD		05-14	05-14	1	31-JUL-82
27	XOUT	.CMD		05-15	05-16	2	31-JUL-82
28	PUTLDR	.CMD		05-17	05-17	1	31-JUL-82
29	NEWDISK	.CMD		05-18	05-1D	6	31-JUL-82
30	NEWBUBL	.CMD		05-1E	05-20	3	31-JUL-82
31	DSP	.CMD		06-01	06-04	4	31-JUL-82
32	SAVE	.LOW		06-05	06-06	2	31-JUL-82
33	FLEXFM8	.SYS		06-07	07-08	34	31-JUL-82

UCS（ユーティリティコマンドセット）

実際にFLEXを動かすためのコマンドが，UCSである。ファイルをコピーしたり，エディットしたり，デリートしたりするユーティリティで，FLEXの特色の一つに，大半のコマンドがディスク上にあるという点があげられる。メモリレジデントコマンド（メモリ上に常駐しているコマンド）は，GET，MONの二つだけである（FM-8用は，さらにOFGET，SETが拡張されている）。これは，コマンドを自由に（OSの変更なしに）拡張できる利点がある反面，ディスケット上にUCSがない場合まったく何もできないことにもなる（ディレクトリも見ることはできない）。この点は，CP/Mユーザーからよく指摘されるが特に欠点ともいえないだろう。名称としては，GET，MONを，メモリレジデントコマンド（CP/Mではビルトインコマンド），その他をディスクレジデントコマンド（CP/Mではトランジェントコマンド）という。

ここまでは，FLEXの内部構造を紹介した。次に，具体的な使い方を説明しよう。

例として，DOSのところで示したプログラムをアセンブルして，コマンドファイルにするまでを説明する。

FM-8で，FLEXを動かすには，ディップスイッチを，図7のように設定する。

電源を入れると，数秒後に，

FM-8　　FLEX　V3.11
DATE（MM, DD, YY）?

という，メッセージが現れる。CP/Mと違い，ファイルには，日付が入るので，DATEの入力が必要である。例えば，「3,18,83」と入れると，FLEXは，「+++」というプロンプトを返す。

CAT

ディスケットに，どんなファイルがあるかは，「CAT」（Catalog）コマンドで見る。ドライブ1の内容を見るには，「CAT 1」とする。FM-8用ディスケットには，図8のような，ファイルが入っている。

EDIT

図3のリストを，まずエディタで作る。ファイル名をTESTとすると，EDIT TEST↵といれる。このエディタは，ラインエディタだが，かなりよくできている。好き嫌いもあるだろうがCP/MのEDよりは，使いやすいと思う。EDIT TEST↵とすると，エディタは，

NEW FILE
1.00=

と表示する。あとは，さっきのプログラムを入れていけばよい。

抜け出すためには，#だけの行を入れる。するとエディタは，#というプロンプトを返す。今，2行目のPUTCHR EQU $CD18を，PUTCHARと打ちまちがえたとする。2行目を表示するには，2↵と打つ。文字の変更はCコマンドを使う。この場合は，C/PUTCHAR/PUTCHR/である。エディットが終了して，FLEXに戻るには，S↵と打つ。エディタのコマンド一覧を図9に示す。

CAT TEST↵と打ってみよう。すると，

TEST.TXTというファイルができている。.TXTは，拡張子で，テキストファイルであることを示す。EDITは，拡張子が省略された場合，.TXTとみなす。TEMP.PRNというファイルを作りたい時には，EDIT TEMP.PRNと，フルネームを打つ。拡張子の例を，図10に示す。

ASMB

次に，今のテキストファイルをアセンブルする。方法は，ASMB TEST↘である。

ASMBは，マクロアセンブラで，条件アセンブリなども備わった強力なアセンブラである(CP/MのASMより，はるかに強力)。ASMB TEST↘の結果エラーがなければ，TEST.BINというバイナリファイルができる。TEST.BINを実行するには，TEST.BIN↘と打ってもよいが，他のユーティリティコマンドと同様に，ファイル名だけで実行可能にするには，TEST.CMDと，コマンドファイルにする必要がある。

RENAME

そのためには，単に，
RENAME TEST.BIN, TEST.CMDと，リネームしてやればよい。この結果，新たにTEST というコマンドが完成する。

RENAMEコマンドを使いたくないときは，アセンブル時に+++ASMB TEST, TEST.CMD↘とすればよい。今，ただTEST↘と打つだけで，自動的にTESTがロードされ実行される（このプログラムは，ASCII文字を全部出力するもの)。

これだけのコマンドを知っていれば，一応FLEXは使えるが，このほかにも，重要なコマンドがいくつかある。

PRINT

FLEXの特色の一つに，スプーリング機能がある。パソコンでは，プリンタに出力している間は，他の仕事が全くできない。しかし，プリンタに対して，CPUが消費される時間はごくわずかで，他の時間は遊んでいる。そこで，タイマIRQによってプリンタを動かし，他のときは別のジョブ（例えばエディトやコンパイル）をバックグラウンドで行えるようにすることを，プリンタスプーリングという。

FLEXでは，スプーリング機能は，標準装備であり，使い方は，

PRINT　ファイル名 ↘

である。さらに驚くことは，いくらでもプリ

図9　EDITコマンドサマリ

コマンド	機能
●環 境 命 令●	
H[EADER] [〈COUNT〉]	〈COUNT〉カラムのヘッダライン表示
NU[MBERS] [OFF/ON]	ラインナンバフラグのON, OFF切り換え
REN[UMBER]	リナンバ
SET〈name〉=‘〈char〉’	特殊記号，シンボルの定義
TAB[〈columns〉]	TABストップのセット
V[ERIFY] [ON/OFF]	VERIFYフラグのON/OFF
X	カーソルコントロール命令
Z[ONE] [C1, C2]	検索エリアをC1～C2に限定
●システム命令●	
LOG	エディットの終了
S[TOP]	エディットの終了
●カレントライン移動命令●	
B[OTTOM]	最後のラインへ
F[IND]〈target〉[〈occurrence〉]	〈target〉を含む行へ
N[EXT]〈target〉[〈occurrence〉]	〈target〉を含む行の次へ
T[OP]	最 初
●編 集 命 令●	
A[PPEND]/〈string〉/[〈target〉]	カレントラインの最後に〈string〉をつけ加えることを〈target〉まで繰り返す
C[HANGE]/〈string 1〉/〈string 2〉/[〈target〉[〈occurrence〉]]	〈string 1〉を〈string 2〉で置き換える
CC[HANGE] [〈column〉]/〈string 1〉/〈string 2〉/[〈target〉[〈occurrence〉]]	〈string 1〉を変更するかどうか選択できるCHANGE
CO[PY] [〈destination-target〉[〈range-target〉]]	カレントラインおよびそれ以降が，〈range-target〉までコピーされ，〈destination-target〉のラインの後に続く
D[ELETE] [〈target〉]	カレントラインから〈target〉まで削除
EXP[AND] [〈target〉]	カレントラインから〈target〉まで，TAB記号を有効化
I[NSERT]	インプットモードへ
MO[VE] [〈destination-target〉[〈range-target〉]]	カレントラインから〈range…〉までを〈dest…〉へ移動
O[VERLAY] [〈delimiter〉]	重複ライン入力
P[RINT] [〈target〉]	〈target〉までプリント
R[EPLACE] [〈target〉]	〈target〉まで削除して入力モードへ
=　〈text〉	カレントラインと〈text〉を入れ換える
(CR)	カレントラインの表示
●テープ命令●	
GAP	null出力
READ	テープよりロード
SAVE	テープへセーブ
W[RITE] [〈target〉]	カレントラインから〈target〉までセーブ

図10

拡張子	意味
.TXT	テキストファイル
.BIN	バイナリファイル
.CMD	コマンドファイル
.BAK	バックアップファイル
⋮ など	

ント待ち行列に入れられることである。

多くのファイルをプリント待ち行列に入れた場合，今どのファイルがキュー（待ち行列）に入っているか，わからなくなる場合があるので，キューの内容を示すQCHECKというコマンドが用意されている。このQCHECKは，プロセスをKillしたりすることもできる。この機能は，パソコンとしてはかなりのハイレベルといえる。このようなバックグランドをさらに進めると，マルチタスクとなる。

O（オー）

O（オー）は，ユーティリティコマンドなどの出力をファイルに変更する。たとえば，CATの出力をcatalog.outというファイルに書きたい場合，

　　O，catalog，cat

となる。この機能は，アセンブラを使う場合有用で，

　　O，test.lst，asmb，test

とすると，CP/Mのアセンブラのリスティングファイルのような，test.lstファイルを作ることができる。Oと同様に出力をプリンタにするPやOとは逆に，キーボードからの入力のかわりに，ファイルから入力するコマンドIもある。このように，標準入出力を自由に変更できる機能をI/O Redirectionというが，FLEXでは，このI/O Redirectionの初歩的な機能も備わっていると言えそうだ（UNIX™などでは，完璧）。

EXEC

CP/MのSUBMITと同じ機能。繰り返し行う一連のコマンドを，ファイルに書いておいて，

　　EXEC　ファイル名

とすると，キーから入力したのと同じ効果が得られる。

STARTUP

STARTUPは，UCSではないが，FLEXシステムがブートした時に，EXECされるファイルである。ブート時に，CATを取りたい場合，STARTUP.TXTというファイルの中にCATと書いておく。すると，ブートするたびに，カタログがとれる。

SET

SETもUCSに入っていない。FM-8用につけ加えられたメモリレジデントコマンドである。SETは，FM-8のターミナルとしての環境を変更する。例えば背景色を青にしたければ，

　　SET　BC ＝ 1 ↘
　　　　バックカラー

図11　UCSサマリ

コマンド	機能
APPEND,〈file spec〉[,〈file list〉],〈file spec〉	ファイルのアペンド
ASN [,W=〈drive〉] [,S=〈drive〉]	ワークドライブ，システムドライブの指定
BUILD,〈file spec〉	簡易エディタ
DATE, [,〈month, day, year〉]	日付の表示，変更
CAT [,〈drive list〉] [,〈match list〉]	ファイルの表示
COPY,〈file spec〉,〈file spec〉etc	ファイルのコピー
DELETE,〈file spec〉[,〈file list〉]	ファイルの抹消
EXEC,〈file spec〉	ファイルの中のコマンド列の実行
I,〈file spec〉,〈command〉	入力をファイルから行う
JUMP,〈hex address〉	〈hex address〉へジャンプ
LINK,〈file spec〉	FLEXオペレーティングシステムを〈file spec〉に割り当てる
LIST,〈file spec〉[,〈line range〉] [,+(options)]	テキストファイルのリスト
NEW DISK,〈drive〉	フォーマット
O,〈file spec〉,〈command〉	出力をファイルへ
P,〈command〉	出力をプリンタへ
PRINT,〈file spec〉[,+〈repeat #〉]	ファイルをプリンタ（キュー）へ
PROT,〈file spec〉[,(option list)]	ファイルのプロテクト
Q CHECK	プリントキューの表示
RENAME,〈file spec 1〉,〈file spec 2〉	ファイル名のリネーム
SAVE,〈file spec〉,〈begin adr〉,〈end adr〉[,〈trans adr〉]	メモリ内容のセーブ
SAVE.LOW	〃
TTYSET [,〈parameter list〉]	ターミナルの機能の変更
VERIFY [,ON/OFF]	ライト時のベリファイの有無
XOUT [,〈drive spec〉]	OUTファイルの抹消

とする。このように，Width，ファンクションキー，文字の色などを，自由に変更する機能を，SET コマンドはもっている。

UCS は，かなり強力で，必要と思われるものは，すべて備わっていると言える。

図11に，UCS のサマリを示す。

FLEXのソフトウェア

次に，FLEX ベースのソフトを見てみよう。

DEBUG PACKAGE

FLEXであきれる(？)のは，デバッガがついていないことだ。あれだけ強力なマクロアセンブラがついていて，あんなに使い勝手のよいエディタがあって，デバッガがない！というのは片手落ちの気がする。

開発ツールとしてFLEX を使う場合，アセンブルして，動かして，「あ！暴走！」また，エディットしなおして，アセンブルして，動かして「あ！暴走！」の繰り返しは極めて非効率である。やはりデバッガは必需品であろう(CP/MではDDTが標準装備)。ということで，このDEBUG PACKAGE (別売)だが，これはDDT よりもはるかに強力だ。ブレークポイント一つとってみても，DDT ではあるアドレスをCPUがインストラクションとしてアクセスしたときに停止するが，FLEXのDEBUG PACKAGEは，8つの異なった状態の一つまたはその組み合わせでブレークすることができる。また，命令実行の速度を調整したり，さらに驚くことには，実行されてしまったインストラクションを(256まで)さかのぼってチェックできる。またメモリのプロテクションも可能で，ICE 顔まけの機能が備わっている（もちろん，アセンブル，ディスアセンブル，トレースもできる)。

FLEX を買う人は，ぜひこのDEBUG PACKAGE も一緒に買うことをおすすめする。

68000 CROSS ASSEMBLER

MC68000用のクロスアセンブラである。このアセンブラも，ASMB と同様，条件アセンブル，マクロ機能付きのよくできたアセンブラである。アセンブルの結果S1/S9，S1/S8という，MACS bug 用のオブジェクトを生成する。筆者は，6809FLEXシステムとMC68000システムをシリアルでつないだシステムで使っているが，安価に68000のプログラムが開発でき，たいへん重宝している。クロスソフトのメリットは，なんといっても，今あるシステムで上位機種のソフトが資金0（もっともクロスソフトのお金は必要だが）で開発できる点であろう。68000のちゃんとしたＯＳが出まわるまでのつなぎとしては，最良の方法だと思う。

このほか，図12に示すようなソフトがある。

図12

名称	内容	記録媒体	価格	提供会社
FM-8用 FLEX UTILITIES	各種ユーティリティ	5D	30,000円	ソフトマート㈱
FLEX UTILITIES	36のユーティリティコマンドがセットされている	5D	25,500円	〃
BASIC	最高速のBASICインタプリンタ	5D	25,500円	〃
STANDARD BASIC PRECOMPILER	BASICのプリコンパイラ	5D	17,000円	〃
68000 CROSS ASSEM	68000クロスアセンブラ	5D	100,000円	〃
DEBUG PACKAGE	プログラムを手直しするツール	5D	25,500円	〃
6809 DIS ASSEMBLER	6809逆アセンブラ	5D	18,000円	〃
EXTENDED BASIC	スタンダードBASIC＋拡張機能	5D	34,000円	〃
EXTENDED PRECOMPILER	EXTENDED BASICのプリコンパイラ	5D	17,000円	〃
FLEX DIAGNOSTIC	11の診断パッケージ	5D	25,500円	〃
firm FORTH	OS，モニタ，プロセス制御などの作成に使用	5D	140,000円	〃
PASCAL	標準パスカルに準拠	5D	98,000円	〃
SORT/MERGE PACKAGE	6809ソート/マージパッケージ	5D	34,000円	〃
TEXT PROCESSING SYSTEM	テキスト・エディタと併用することにより，英文ワードプロセッサとなる	D	25,500円	〃
EXCEL-9F(FLEX版)	8Kバイトの機械語モニタ	5D	12,800円	㈱星光電子
FLEX	モトローラ系MPUのOS	5D 8D	58,000円	富士通㈱
t FORTH	FIG(FORTH Interest Group)の拡張版	5D	40,000円	ソフトマート㈱
t FORTH ＋	t FORTHの拡張版	5D	100,000円	〃
FORTRAN 77	FORTRAN	5D	165,000円	〃
Relocating Assembler & Linkage Editor	リンキングローダ		発売予定	〃
Extended Precompiler	6809拡張プリコンパイラ	5D	17,000円	〃
micro C	ソフトウェア言語	5D	50,000円	㈱生活構造研究所 マイクロコミュニケーションズ

（富士通カタログより）

おわりに

FLEX について，まだまだ言いたいことはあるが，このへんで終える。 FM-8(FM-7，FM-11 でも動く）用のＯＳとしては，このFLEXはもっとも手近で使いやすいものと言えるだろう。

F-BASIC中級入門

2

桑原　岳夫

はじめに

FM-8に7/11という２機種の新製品が加わりFMシリーズとしてラインアップされた。特にFM-7はFM-8で改良の余地のある部分がほとんど改良され，コンピュータのクロックがメイン/サブとも2MHzになったことにより，８ビットCPUを使ったマイコンとしては限界に近い性能を発揮している。またFM-8と基本構成が同じためBASICレベルでコンパチビリティを備えている，などなかなか考えている。実際にFM-7はよく売れているらしく，NECも対抗機種としてPC-8001MKIIを出してきた。個人的意見としては，FMシリーズの方が基本的な設計が優れているため，PCシリーズに対してかなり優位な位置にいると思う。しかし，マイコンはソフトウェアが充実してこそ完全になる，ということを考えあわせると両者のソフトウェア蓄積量に大差があり，FMシリーズの苦戦が続くのではないか，とも思われる。

いずれにしろ富士通には，くれぐれも某社の二の舞は演じてほしくない。マイコンは家電製品ではないのだから。

話はかわるが，FM-8用の入門書がいろいろ出版されてきた。けれども金をだしても手に入れたいと思う著作は少ない（もちろん自分のことは棚に上げて話す）。その中でこれはいいなと思ったのは「FM-8操縦法入門」（脇英世著）という入門書だ。すこし高価ではあるが，初心者から中級者まで読んで役に立つ事項が多い。著者自身がユーザーでFM-8の内部構成まで完全に理解している。（FM-8の構成――アーキテクチャ――は当時としては斬新でマニア受けした機械だが，初期のこの機械に関する情報の少ないときに苦労して内部を解析した人たちが多くいた）

脇氏にはこのほかにもブルーバックスに著作があり，よい評価をうけている。このことからもわかるように，マニアにだけではなく，初心者に対する氏の配慮が行き届いた解説書となっている。

さて第１回では「BASIC中級入門」と銘打ったにもかかわらず，DISK BASICの基本的な事項しか書かなかった。ある人から，「中級入門なのに結局はDISKに関することしか書かれていない」というクレームがついたが，確かにそのとおりであり，それだけなら参考書を読んだ方がためになるかもしれない。

これからはできるだけBASICに関することのみではなく，その内部まで踏みこんで書いていきたいと思う。

まず手はじめにシーケンシャルファイルを使ったプログラムについて，

- 文字列のサーチ
- 成績処理に関してのソーティング

の２つの題目を使って話を進めていく。

サーチ

ソフトウェアをつくる過程において，非常に便利なコマンド機能がいろいろある。たとえばFMシリーズのスクリーンエディタなどは，他機種と異なりスクリーンのどの場所を修正しても［Return］キーを押すだけで，すべて修正できるという機能になっている。富士通の大型機の端末のスクリーンエディタもこの機能をもっているが，一度慣れてしまうと非常に便利なものだ。

この項ではシーケンシャルファイルを使用し，BASICで簡単に実現できる機能の例として文字列サーチプログラムを作成する。市販のプログラムで，「音がうるさいのでBEEP音を出している命令のみを消すために，その行番号を探したい」ときや，「ある変数を使用している行番号すべてを知りたい」ことがある。そのような場合，LISTをとりスクリーンとにらめっこしながら一生懸命文字列を探すのはたいへんだ。そんなときにあると便利なプログラムである。

BASICによってはこの命令をもっているものもあるが，F-BASICなどマイクロソフト系のBASICには用意されていないようだ。もし命令の文字列がそのままマイコンのプログラム領域に収納されているならば，文字列サーチは簡単に行えるはずだ。初期のマイコン用BASICはこの，そのまま収納する方法をとっているものが多くあった。これは，LIST命令や入力を実行するためのBASIC本体(機械語で書かれている）が小さく作成できることと，それ自体簡単にできることによる。現在でも小型の一部のマイコンではこの方式を採用している。それに対して最近のマイコンは入力されたプログラムを単語に分解し，ちょうど高校生の頃の古文の授業でやったように，BASIC本体が品詞にまで文を分解していく。命令は小さく圧縮してメモリに収納し，命令以外の単語はそのまま記憶していくようになっている。こうすることによりあらかじめ前処理によって実行時の作業を減らすことができ，あわせてプログラムを収納するメモリの使用量を省くことができる。

マイコンに内蔵されているBASICはすべてといっていいほど，インタプリタ（通訳）系BASICである。これはBASICなどの高級言語の命令体系を実行する仮想コンピュータをつくり，それを機械語でシミュレートして実行する（高級言語というのは高級な言語という価値判断を示すのではなく，最終的に実行する機械語からはなれて抽象化されているものを示す）。

こうすることによって実行時におかしな命令が出た場合，実行を中止したり，実行中に一時的に止めたり，変数の値を変更したりする機能を実現している。

このため機械語でプログラムをつくるときに暴走が起きることが少なく，もし起きたとしてもプログラムの破壊を防ぐことができる。ただしコンピュータでコンピュータの動作をシミュレートしているため，BASICの実行速度は機械語でつくったプログラムの，数百分の1に落ちてしまう。よく「BASICは遅い」といわれる理由だ。

話が横道にそれたが，BASICでは速度の遅さを少しでもカバーするために，現在では圧縮した形――中間言語――でメモリに収納している。そのためLIST命令を実行するときなどはかなり複雑な動作――中間言語から高級言語へ変換する――を行っている。中間言語を実際に見たいときは，

SAVE "SCRN:" (CR)

を実行する。プログラムがリストのときとは，異なった いろいろな文字で画面に表れると思う。この形で収納されているプログラムでは，サーチを行うことが難しいため，ここでは，プログラムのシーケンシャルファイルをフロッピーにつくり，サーチを行ってみよう。

サーチプログラムをリスト1に示す。

まずサーチしたいプログラムのファイルを

LIST "0:TF" (CR)

を実行してつくり，リスト1を実行する。そしてサーチしたい文字列を入力して (CR) キーを押すと，その単語のある行番号を画面に表示する。プリンタに結果を出したいときは，ファイル2のOPEN命令のファイルディスクリプタをSCRN:からLPT0:に変えて使う。このような簡単なプログラムで済んでしまう。

リスト 1

```
1000 '     EX.1  (PROGRAM FOR SEARCH)
1010 '
1020 CLS
1030 LINEINPUT DA$
1040 OPEN"I",1,"0:TF"
1045 OPEN"O",2,"SCRN:"
1050 WHILE EOF(1)=0
1060  LINEINPUT#1,BF$
1070  IF INSTR(BF$,DA$)=0 THEN 1090
1080  PRINT#2,BF$
1090 WEND
1100 END
```

成績処理

ここでは，「0～100点のテストを100人に行い結果を出す。そして得点の高い順に並べかえ，同時に標準偏差および平均を算出する」というプログラムをつくる。

並べかえは一般にソーティング(Sorting)と呼ばれ，コンピュータでデータを処理する場合に必要となることが多い技法である。そのためコンピュータサイエンスの部門でかなり研究され，いろいろな方法が開発されている。例をあげると，

(1) バブルソート
(2) クイックソート
(3) ヒープソート

などがある。名前を聞いただけでは内容の見当はつけにくいと思う。(1)は簡単なプログラムで済むためにデータが少ないとき有効だが，データが増えるにつれソーティング速度が遅くなり，(2)，(3)を使わなければまったく実用に供さない。

ここではデータの量も少なく，マイコン自体データを多量にメインメモリ上にもてないため，(1)の改良型を使用する。

バブルソートとは「すべてのデータに渡り，隣り同士のデータの大小を比較し，交換を行い，交換が行われなくなったときデータの並べかえが終了したとする」というものである。ピンとこないならトランプを並べて実際に動作をやってみると理解しやすくなる。

この方法の欠点はデータの並び方により並べかえ速度が変わることと，交換が隣り同士としか行われないために，データが多くなるとどうしても速度が落ちることなどだ。そこで初めに交換する数の距離を大きくとり，だんだん小さくしていけば並べかえの効率があがる。実際のプログラムをリスト2に示す。

ここでは100人のデータについて処理を行ったが，1000人以上の多量のデータを並べかえる場合はどうであろうか。

もし100人分ほどの配列しか内部にとれない場合は，ソーティングの操作ができない。けれどもDISKがあれば1000人を十等分して100人ずつのファイルに分割し，それぞれをソーティングした後に一つのファイルに順位どおりまとめればよい。前回で，

How many disk files ?

とDISK BASIC起動時に表示がでる，と書いたが，ここでオープンするファイルの数を指定できることが生きてくる。考え方としては簡単だが，ディスクがあればこその技法だ。

リスト 2

```
1000 '  EX2     (PROGRAM FOR SORT)
1010 '
1020 DI=0
1030 WHILE 2^DI<XNUM
1040   DI=DI+1
1050 WEND
1060 DI=DI-1
1070 WHILE DI>=0
1080    DE=2^DI:DF=1
1090   FOR I=1 TO XNUM-DE
1100           DA=SU(I):DB=SU(I+DE)
1110           IF DA<DB THEN SU(I)=DB:SU(I+DE)=DA:SWAP BAN(I),BAN(I+DE):DF=0
1120   NEXT I
1130 DI=DI-DF
1140 WEND
```

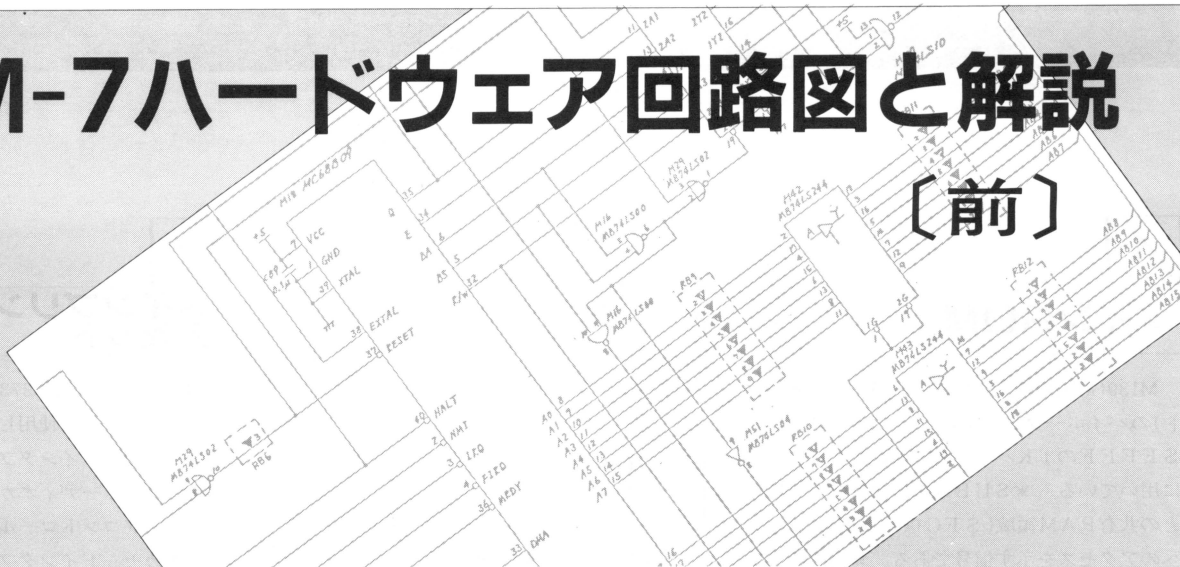
ソーティングについての詳細は，プログラムの入門書なら必ず書いてあるので，それを参照してほしい。

特に（2），（3）の技法は再帰的に表現しやすいため，またそう表現した方がプログラムに簡潔に表せるため，Pascalなどの入門書には再帰的プログラミングの例題として必ず載っている。（ほとんどのBASICは変数名の多重定義を許していないので，再帰的表現が苦手である。）

ともあれ初めに示した例題のプログラムをリスト3に示す。初めの方が100人分の成績データをファイルとしてつくる部分で，後半によりファイルの入力，並べかえ，出力を行っている。ここでは記録のかたちとして整数を用いているが，浮動小数点を用いればそれに対応してデータを読み出せばよい。プログラム自体は簡単なものだから解読してみてほしい。これは土台だから自由に修正，改良していろいろ試してみるといいだろう。ファイルの結合なども自分でいろいろ試みてほしい。

リスト 3

```
1000 '     EX.3  (PROGRAM FOR EXERCISE)
1010 '
1020 CLS
1030 OPEN"O",1,"0:TENSU"
1040 '
1050 '    GENERATE DATA
1060 '
1062 DEFINT A-Z
1065 XNUM=100
1070 FOR I=1 TO XNUM
1080    TEN=0
1090    FOR J=1 TO 10
1100         TEN=TEN+RND
1110    NEXT J
1120    TEN=INT(TEN*10)
1130    PRINT#1,TEN
1140 NEXT I
1150 CLOSE#1
1160 '
1170 STOP
1180 '
1190 '    INPUT DATA FROM DISK
1200 '
1210 DEFINT A-Z
1220 OPEN"I",1,"0:TENSU"
1230 XNUM=100
1240 DIM SU(XNUM),BAN(XNUM)
1250 '
1260 FOR I=1 TO XNUM
1270   INPUT#1,SU(I)
1280   BAN(I)=I
1290 NEXT I
1300 CLOSE#1
1310 '
1320 '    SORTING
1330 '
1340 DI=0
1350 WHILE 2^DI<XNUM
1360   DI=DI+1
1370 WEND
1380 DI=DI-1
1390 WHILE DI>=0
1400    DE=2^DI:DF=1
1410   FOR J=1 TO XNUM-DE
1420           DA=SU(J):DB=SU(J+DE)
1430           IF DA<DB THEN SU(J)=DB:SU(J+DE)=DA:SWAP BAN(J),BAN(J+DE):DF=0
1440   NEXT
1450 DI=DI-DF
1460 WEND
1470 '
1480 '    PRINT OUT DATA
1490 '
1500 OPEN"O",1,"SCRN:"
1510 FOR I=1 TO XNUM
1520   PRINT#1,USING"### ";BAN(I);:PRINT#1,SU(I)
1530 NEXT I
1540 END
```

FM-7ハードウェア回路図と解説〔前〕

1 メインCPU

メインCPU（MBL68B09）のアドレスバス（A０～A15），データバス（D０～D７），CPU制御信号（Q，E，BA，BS，R／W）は，それぞれLSTTLによるバスバッファ（MB74LS244，MB74LS245）にてバッファリングされてから本体内部の各回路に接続されている。アドレスバス，データバス，Q，E，R／Wの各信号線は，CPUがバス解放状態（BA＝1，BS＝1）のときに，ハイインピーダンス状態になる。

CPUのクロックは外部から供給される。またCPUのQ，Eクロック信号からZ80CPU（オプションで使用される）のクロックであるZ800が作成される。BA信号とBS信号とのNAND出力（＊REFGRNT）を，メインDRAMのリフレッシュ許可信号として出力しているが，この信号はZ80CPUの動作中（Z80W＝L）のときには出力されない。

リセット回路は電源投入時のパワーONリセット回路とリセットスイッチによるマニュアルリセット回路の２つが並列に（NAND）接続されている。

2 クロック制御

M83（MB74LS158）にて，メインCPU，サブCPUの動作クロックを切り換える。

SCLKNMIは約20ms周期のクロック信号であり，動作クロック切り換えスイッチの内容をラッチする。SCLK１は8MHzの，SCLK２は4MHzのクロック信号である。

また，メインCPUの動作クロックはMCPUCLK，サブCPUの動作クロックはSCPUCLKである。M38（MB74LS367A）のG２側のバッファゲートは，＄FD00番地の入力ポート（bit0）となっており，CPUの動作クロック周波数の読み込み用として使用している。

M38（MB74LS367A）のG１側のバッファゲートは＄FD03番地の入力ポート（bit0～bit3）となっており，IRQ割り込み原因を読み込むために使用する。＊KEYINはキー入力割り込み信号，＊LPINTはプリンタ割り込み信号である。M50（MB74LS74A）はタイマ割り込み（２ms間隔）の制御に用いられる。＊TMMASKはタイマ割り込みのマスク信号，２MSはM61（MB84040B）にて分周された2ms周期のタイマクロック信号，IRQCLRはCPUが＄FD03番地をリードすることにより発生するIRQのアクノリッジ信号である。

M73（SN74LS393），M50（MB74LS74A）はCPUが2MHzで動作しているときに，I／O領域（＄FD00～＄FDFF）およびブートROMのアクセス時に，1/4サイクルのメモリレディをかけるための回路である。＊IOSは，I／O領域のアクセスを，＊BTRDYはブートROMのアクセスを示す信号線である。＊MRDYはメインCPUのメモリレディ（MRDY）端子に接続される。

3 メインI/Oアドレス

メインCPUのI／O領域（＄FD00～＄FDFF）のアドレスデコード回路である。

＊IOSはアドレスバスの上位8ビット（AB８～AB15）をデコードしたI／O領域（＄FD00～＄FDFF）へのアクセスを示す信号線である。＊FD0Xはアドレスの上位12ビット（AB４～AB15）までをデコードした＄FD00～＄FD0F番地へのアクセスを示す信号線だ。

M40（MB74LS138）はライト信号のデコードを行う。＊WFD00～＊WFD03は，それぞれ＄FD00～＄FD03番地へのアクセスを示す信号である。＊WTQEはCPUのQ，EクロックのNAND（否定論理積）と，R／W信号とのOR（論理和）をとった信号だ。

＊WFD05は＄FD00～＄FD03とは少し異なったタイミングで出力するために，別回路でデコードしている。M52（MB74LS138）はリード信号のデコードを行う。＊RFD00～＊RFD05はそれぞれ＄FD00～＄FD05番地へのアクセスを示す信号である。

＊PLTREGは，パレットレジスタ（＄FD38～＄FD3F）へのアクセスを示す信号である。M93（MB74LS138）は，マルチページ制御レジスタ（＄FD37）への書き込みを示す＊WFD37信号をデコードする。

4

メインROM

M139(MB7053)は2Kビット(512×4ビット)のバイポーラPROMであり，＄FC00～＄FFFFの1Kバイトのアドレスデコードに用いている。＊SUBSELはサブCPUとの共有RAM領域(＄FC00～＄FC7F)へのアクセスを示す信号である。＊MIOSはI／O領域（＄FD00～＄FDFF）の中で，本体内部で使用しているI／O番地を除いた残りの番地に対するアクセスを示す信号で，拡張バスポートのデータバッファの制御に用いる。＊BTROMはブートROM(＄FE00～＄FFEF）へのアクセスを示す信号である。＊RAMIHB1はメインDRAMをインヒビットするための信号である。

M152（MB8516）はブートROMとして用いられる16Kビット（2Kバイト）のEPROMである。

M151（MB83256）はBASICインタプリタの格納された256Kビット（32Kバイト）のCMOSマスクROMである。M132（MB74LS74A）は裏RAMの制御に用いられているDフリップフロップである。また＊RAMIHB2はベーシックROM（MB83256)がアクセスされているときに，メインDRAMをインヒビットするための信号である。

5

メインRAM

メインRAMにはリフレッシュ機能付きの64KDRAMであるMB8265を8個用いて，64Kバイトのメモリを構成している。

DL3は総遅延時間200nsのディレイラインであり，DRAMのタイミング制御に用いている。MB8265の$\overline{CAS}$, $\overline{RAS}$, $\overline{RFSH}$などの信号は，このディレイラインによって作成される。RAMIHB1，RAMIHB2はメインRAMのインヒビット信号であり，＊REFGRNTはDRAMのリフレッシュ信号である。

M149，M137（MB74LS158）はアドレスバスのマルチプレクサとして用いられている。M119，M130（MB74LS04）およびM150（MB74LS240）はデータバスバッファとして用いる。DRAMのデータバスは負論理となっている。

6

メインブザー

M19（MB14415）はブザーコントロール，DRAMのリフレッシュ制御，タイマクロックの発生，などの機能を持った200ゲートのバイポーラゲートアレイ（セミカスタムIC）である。SP端子は約830Hzのブザー出力信号端子であり，20MS端子はインタバルタイマ用の20ms周期のクロック出力信号である。DMA端子はDRAMのリフレッシュのタイミング発生に用いられる約13μs周期のメインCPUに対するバス要求信号で，メインCPUのDMA端子に直接接続される。BUZZER信号はサブCPUからのブザー信号であり，0.3MHz信号は4.9152MHzのクロックを16分周した約0.3MHzのクロック信号である。＊REFGRNT信号は＊DMA信号がメインCPUに受け付けられたことを示す信号で，＊DMA信号の解除に使用される。

M72（MB74LS367A）のG1側のバッファゲートは，＄FD04番地の入力ポートとなっており，G2側のバッファゲートは＄FD05番地の入力ポートとなっている。＊BREAKはブレークキーからのFIRQ割り込み要求信号である。＊EXTDETは拡張バスポートからのイクスターナルデテクト（拡張検出）信号である。BUSY信号はサブCPUからのビジィ信号であり，メインCPUとサブCPUとのインタフェースに用いられる。

M45（MB74LS74A）はサブCPUからのアテンション割り込み信号である＊ATTENT信号の制御を行うDフリップフロップである。

7

メインプリンタ，カセット

M10（SN74LS273）は＄FD00番地の出力ポートとして使用している。bit0はオーディオカセットインタフェースへの出力信号に，bit1はオーディオカセットインタフェースのモータコントロールに用いられる。オーディオカセットインタフェースの入力はM35（MB3614）によって波形整形されて，＄FD02番地の入力ポートであるM1（MB74LS244）のbit7に接続される。

M10（SN74LS273）のbit6はプリンタへのストローブ信号線に用いており，M15（MB74LS74A）による内部BUSY信号の作成にも用いる。M2（SN74LS273）は＄FD01番地の出力ポートであり，プリンタへのデータ出力に用いる。M1（MB74LS244）は＄FD02番地の入力ポートであり，プリンタからのステータス信号の入力ポートとして使用する。コネクタ（CN2）の14番ピンはプリンタからのビジィ信号だが，これはM15（MB74LS74A）にて作成される内部BUSY信号とのOR（論理和）をとってから入力している。コネクタ（CN2）の16番ピンはプリンタからのアクノリッジ信号であり，M15（MB74LS74A）によるプリンタ割り込み信号（＊LPINT）の作成にも使用される。＊LPMASKはプリンンタ割り込みマスク信号である。

M9（MB74LS175）は＄FD05番地の出力ポートであり，Z80CPU（オプション）の制御とサブCPUとのインタフェースをとるための＊CANCEL信号，＊SUBHALTREQ信号の出力に用いる。

8

共有メモリ

M122（MB8128）はメインCPUとサブCPUとのデータのやりとりをするための共有RAMとして用いられている2KバイトのNMOSスタティックRAMだ。

＊SHALTAC信号はサブCPUのホルト(HALT)が受け付けられたことを示すもので，この信号がLOWになっているときは，サブCPUのアドレスバス(SADDRBUS)，データバス（SDATABUS）はハイインピーダンス状態になっている。

＊SUBSELはメインCPUの共有RAMに対するデコード信号であり，＊SSMEMはサブCPUの共有RAMに対するデコード信号である。共有RAMはメインCPUからは＄FC80～＄FCFFの128バイトに，サブCPUからは＄D000～＄D7FFの1Kバイトにデコードされる。

9 サブCPU

サブCPU（MBL68B09）のアドレスバス（A0～A15），データバス（D0～D7），CPU制御信号（Q，E，BA，BS，R／W）はそれぞれLSTTLによるバスバッファ(MB74LS245，MB74LS244，MB74LS367A）にてバッファリングされてから本体内部の各回路に接続されている。データバス，アドレスバスはCPUがバスを使用しないとき（BA＝H）にハイインピーダンス状態になる。

10 サブアドレスコード

＊SROMSEL，＊SROMD，＊SRAM1CS，＊SRAM2CS，＊SSMEMはサブCPUに接続されているROM，RAM（VRAMは除く）のアドレスデコード信号である。M87，M96（MB74LS138）はサブCPUのI／Oのデコードに用いている。＊BUZZERはブザー音発生信号であり，＊ATTENTはメインCPUへのアテンション割り込み信号である。

M95（MB74LS139）はVRAMのアドレスデコードを行っている。またM66（MB74LS32）はアクティブページの制御を行っている。＊VPAGE1～＊VPAGE3はアクティブページの選択信号であり，＊SDRAMV1～SDRAMV3はサブCPUからのVRAMへのCAS信号の選択信号である。

DL1は総遅延時間200nsのディレイラインであり，サブCPUがVRAMをアクセスするときのタイミング信号の発生に用いられる。

11 サブROM/RAM

M141，M123（MB8128）は2KバイトのNMOSスタティックRAMであり，サブCPUのコンソールバッファとしてCRT画面に表示される文字の文字コード，およびそのアトリビュートコードを記憶するために用いる。アドレスはM141が＊SRAM1CSにて＄C800～＄CFFFの2Kバイトに，M123が＊SRAM2CSにて＄C000～＄C7FFの2Kバイトにデコードされる。

M154（MBM2764-20）はCRTモニタが格納された64Kビット(8Kバイト)のEPROMである。アドレスは＊SROMSELにて＄E000～＄FFFFの8Kバイトにデコードされる。

M153（MBM2732A-20）はCRT表示文字用のキャラクタジェネレータとして用いられている32Kビット(4Kバイト)のEPROMだが，4Kバイトのうち半分の2Kバイトのみ使用している。アドレスは＊SROMDにて＄D800～＄DFFFの2Kバイトにデコードされる。

12 サブREG/FLAG

M46（SN74LS273）はアクティブVRAM，ディスプレイVRAMの制御を行うためのマルチページレジスタとして用いられている。このレジスタは＊WFD37によってメインCPUにアクセスされる。

M56，M44，M45（MB74LS74A）はサブCPUのフラグの制御に用いる。各フラグの定義は次のとおりである。

＊SVDOFF：CRTの表示を停止するための信号

INS：キーボードのインサートモード表示用LEDの点灯信号

＊SUBIRQ：メインCPUからの＊CANCEL信号に対するサブCPUへのIRQ割り込み信号

＊BUSY：サブCPUがビジィ状態にあることを示す信号線

＊SHALT：サブCPUをホルトするための信号で，CPUのHALT端子に接続される

＊SHALTAC：サブCPUのホルトが成立したことを示す信号

13 サブCRT CNTRL

M124（MB60H010）は高速型2000ゲートCMOSゲートアレイ（セミカスタムLSI）であるCRTコントロール用のICである。このICはCRTへの表示タイミングクロックの作成，同期信号の作成，VRAMのアドレス信号の出力，スクロール処理などの機能を64ピンパッケージに収めてある。

14 サブCRTアドレス

VRAMのタイミング信号の作成回路である。M59（MB462）はサブCPUからのタイミング信号とM124（MB60H010）からのタイミング信号の選択を行っている。DL2は総遅延時間50nsのディレイラインであり，VRAM1のアドレスの切り換え信号（SADRSEL）とVRAM用DRAM（MB8116H）のCASのタイミング作成に用いる。

メインボードの回路図は，このあと，⑮サブCRT　RAM，⑯サブCRTインタフェース，⑰キーインタフェース，⑱拡張バスバッファ，⑲コネクタ（拡張，Z80），⑳コネクタ（PSG），㉑コネクタ（RS-232C，漢字ROM），㉒メインPSG，㉓電源，と続くが，これらは次号に掲載する。

FM-7ブロック図

AC.100V
電源回路
+5V
+12V
-12V
GND
リセットSW
リセット回路
RESET
64Kバイト RAM
メイン・メモリ
MB8265 ×8
RAS CAS WE
RFSH
制御回路
REFCK
BASIC
32Kバイト ROM
MB83256
DIP
SW
ブートローダ
480バイト ROM
MB8516
ラインプリンタ
コネクタ
ラインプリンタ
インターフェース回路
オーディオカセット
コネクタ
MIC
PHONE
REMOTE
オーディオカセット
インターフェース回路
MB3614
内部SP
外部SP
スピーカ出力回路
PSG回路
AY-3-8913
BEEP音
発生回路
I/O拡張スロット
拡張
バスコネクタ
拡張
バスバッファ
R/W
E
EQ
IRQ
EXTAL
NMI
FIRQ
割込み
制御回路
メイン
CPU
MBL
68B09
バス・バッファ回路
アドレス・バス
データ・バス
コントロール・バス
HALT
CPU
制御回路
Z-80カード
アクティブページ
制御回路
HALT
制御回路
RAM
制御回路
メイン サブCPU
共有メモリ(128B)
サブ RAM(5KB)
MB8128 ×3
クロック
コントロール
回路
クロック
発生
回路
4.9152MHz
DIP
SW
サブ
CPU
MBL
68B09
バス バッファ回路
CRT モニタ
10Kバイト ROM
MB8364(モニタ)
MB8516(CG)
TV走査信号
発生回路
16.128MHz
RAS CAS WE
制御回路
RAS
CAS
WE
ビデオ RAM
48K バイト
MB8116×24
R
G
B
マルチページ
パレット
制御回路
ビデオ
信号
発生回路
コンポジット
ビデオ出力
コネクタ
RGB出力
コネクタ
スピーカ出力
INS LED表示回路
INS LED
カナ
LED
CAP
LED
キーボード データ
制御回路
キーボード インターフェース回路
4ビット ワンチップ マイコン
MB88401
キー
ストローブ
キーボード
CLOCK
STOP
*オプション

①図　FM-7メインCPU

②図　FM-7クロック制御

③図 FM-7メインI/Oアドレス

④図　FM-7メインROM

⑤図 FM-7メインRAM

⑥図 FM-7メインブザー

⑦図　FM-7メインプリンタ，カセット

⑨図　FM-7サブCPU

M85 MC68B09
M74 MB74LS367A
M109 MB74LS244
M110 MB74LS244
M75 MB74LS245
M65 MB74LS08
M63 MB74LS00
M64 MB74LS32
MB74LS04 M76
RB13
RB14
RB15
RB16
RB17
RB18
(1-C11) *RESETB
*SCRESET (12-B1)(12-D1)
(2-C4)SCPUCLK
(6-C11)*SCLKNMI
(12-D4) *SUBIRQ
(17-F7) *KSTROBE
(12-B11) *SHALT
*SWTOE (8-E1)(11-F1)
*SQANDE (10-E1)
*SRDOE (8-E1) (11-F1)
SQB (10-A1)(10-C7)
SEB (10-B1)(10-D7)
*SHALTST (12-B9)
SRWB (12-B1)(12-B5) (10-C7)
*SRWB (15-H1)
SBA (10-B1)
SDATABUS (8-A11) (11-F11) (13-E1) (15-A1) (17-A1)
SADDRBUS (8-A1)(10-A1)(11-A1)(13-A1)

⑩図　FM-7サブアドレスデコード

⑪図　FM-7サブROM/RAM

⑫図　FM-7サブREG/FLAG

⑬図　FM-7サブCRT CNTRL

さがしものゲーム

平賀富佐子

わたし，女子大生，ゲームをつくってみました。あまり整理されたプログラムとは言えません。ごめんなさい。

ピンクのハートを動かして，石に隠れた果物を次々に探してください。CRTの左上にりんごが3個出ていたら，3個りんごを見つければ，あなたの勝ちです。

なーんだ，単純すぎてバカみたいだ，ですって？でもそれだけじゃないんです。

赤いスペードが，あなたのハートを追ってきます。それに食べられると，死んじゃうんです。なーんだ，やっぱり単純だ，なんて言わないで。

赤いスペードが通ったあとは，青いスペードの壁ができます。ピンクのハートは，その壁を越えられません。どうだ，少しは困ってきただろー。壁に囲まれると，ハートは逃げられません。追りくるキョーフ。ハートは3個食べられると終わりです。

それから帽子をかぶった黄色いヘビは，問題の果物と違っていたときに，にょろにょろ前に進みます。左端まで行くと時間切れ。もぐらが出てくると，得点が減るから気をつけてくださいね。

@のしるしが出ていると，そこからワープで逃げられます。でも，どこに飛び出すかわかりません（もっと危いところに飛び出して食べられたらごめんなさい)。@のしるしは，はじめのうちは出ません。

単純そうにみえて，結構たいへんなんです。ゲームの終わりに，得点によって5種類のメッセージが出ます。

赤いハートは8(↑)4(←)6(→)2(↓)で動かしてください。ヘビより上へ動かしたりすると，2(↓)のキーが効かなくなるので注意してください。

```
70 DEFINT A-X,Z
80 DIM B1(450),A1(102),A2(66),A3(91),A4(127),A5(55),A6(159)
90 DIM C(16),D(16),E(32),Z1(50),G(16):P#=0:SBL=0:MEN=1:S1=0
100 WIDTH 40,25:COLOR7:CLS
110 LINE(5,5)-(34,19),"*",6,BF:LINE(8,8)-(31,16)," ",,BF:SYMBOL (208,80),"ｻ  ｶﾞ
 ｼ  ﾓ  ﾉ",2,2,7,0,PSET:SYMBOL (256,104),"ｹﾞ  -  ﾑ",2,2,7,0,PSET:FOR I=1 TO 1000:
NEXT I:CLS
120 '
130 '    **  ｶﾞﾒﾝ ﾉ ｷｬﾗｸﾀｰ   **
140 '
150 LOCATE 11,3:PRINT"ｷｬﾗｸﾀｰ  ﾉ  ｼｮｳｶｲ"
160 RESTORE 2400:Z1=49:Z2=6:GOSUB1170:PAINT(170,58),6,6:Z1=1:Z2=0:GOSUB1170
170 Z1=4:Z2=3:GOSUB1170:PAINT(178,54),3,3:Z1=17:Z2=2:GOSUB1170:Z1=22:Z2=5:GOSUB1
170
180 PAINT(414,66),1,5:PAINT(424,76),1,5:PAINT(411,76),1,5:CIRCLE(411,69),6,7,,0,
1,F:CIRCLE(421,69),6,7,,0,1,F
190 Z1=2:Z2=7:GOSUB1170:Z1=2:Z2=0:GOSUB1170:Z1=4:Z2=7:GOSUB1170:PAINT(412,74),7,
7
200 Z1=1:Z2=2:GOSUB1170:Z1=4:Z2=7:GOSUB1170
210 FOR I=1 TO 6:READ Y1,Y2,Y3,Y4:CIRCLE(Y1,Y2),5,7,.5,Y3,Y4:NEXT I
220 PAINT(100,114),0,7,0:Z1=3:Z2=7:GOSUB1170
230 FOR I=1 TO 32 STEP 2:LINE(201+I,116)-(201+I,128),PSET,2:LINE(201+I+1,116)-(2
01+I+1,128),PSET,6:NEXT I
240 CIRCLE(219,122),14,7,.41:CIRCLE(219,119),3,0,,0,1:Z1=5:Z2=0:GOSUB1170:PAINT(
205,116),0,7:CIRCLE(322,122),11,7,.5,.67,.33:CIRCLE(311,122),11,7,.5,.17,.88
250 CIRCLE(317,119),7,7,,.05,.45:PAINT(317,119),2,7:LINE(316,119)-(318,115),PSET
,4,BF:CIRCLE(423,122),18,4,.4,0,1,F:Z1=2:Z2=7:GOSUB1170:Z1=9:Z2=0:GOSUB1170
260 FOR I=1 TO 32 STEP 2:LINE(499+I,118)-(499+I,129),PSET,2:NEXT I
270 CIRCLE(515,125),13,7,.3,.05,.45:Z1=10:Z2=7:GOSUB1170:CIRCLE(515,128),12,7,.3
,.6,.9:PAINT(515,127),1,7:PAINT(515,127),2,7:LINE(498,117)-(532,130),PSET,1,B
280 PAINT(499,118),1,7:PAINT(499,118),0,7
290 GET@A (148,53)-(246,75),B1,G:GET@A (100,111)-(132,126),A1,G:GET@A (204,117)-
(233,127),A2,G:GET@A (299,115)-(334,127),A3,G
300 GET@A (404,113)-(442,129),A4,G:GET@A (502,119)-(529,128),A5,G:GET@A (391,64)
-(442,79),A6,G
310 LOCATE 4,20:PRINT"ｹﾞｰﾑ ﾉ ｾﾂﾒｲ ｦ ｼﾏｽｶ?  YES ﾅﾗ <Y>":LOCATE4,22:PRINT"NO ﾅﾗ <N
> ｦ ｵｼﾃｸﾀﾞｻｲ.";
320 Z$=INKEY$:IF Z$="" THEN 320
330 IF Z$="N" OR Z$="n" THEN PRINT"N":FOR I=1 TO 600:NEXT I:GOTO 400
340 IF Z$="Y" OR Z$="y" THEN PRINT"Y":FOR I=1 TO 600:NEXT I:GOSUB 1220
350 GOTO 320
360 '
370 '    **  ﾓﾝﾀﾞｲ   **
380 '
390 IF MEN=13 THEN MEN=1
400 CLS:LINE(0,4)-(39,24),"■",0,B:IF MEN=1 OR MEN=4 OR MEN=7 OR MEN=10 THEN REST
ORE2770:MAL=14:COLOR6
```

```
410 IF MEN=2 OR MEN=5 OR MEN=8 OR MEN=11 THEN RESTORE2780:MAL=15:COLOR5
420 IF MEN=3 OR MEN=6 OR MEN=9 OR MEN=12 THEN RESTORE2790:MAL=16:COLOR4
430 FOR I=1 TO MAL:READ C(I),D(I):LOCATE INT(C(I)/16),INT(D(I)/8):PRINT"●":NEXT
I
440 COLOR7:RANDOMIZE(VAL(RIGHT$(TIME$,2))):A=INT(RND(2)*5)+1:B=INT(RND(10)*5)+2
450 FOR I=0 TO B-1
460 IF A=1 THEN PUT@A(227-I*38,16)-(259-I*38,31),A1%,PSET
470 IF A=2 THEN PUT@A(230-I*38,16)-(259-I*38,26),A2%,PSET
480 IF A=3 THEN PUT@A(224-I*38,16)-(259-I*38,28),A3%,PSET
490 IF A=4 THEN PUT@A(221-I*38,16)-(259-I*38,32),A4%,PSET
500 IF A=5 THEN PUT@A(232-I*38,16)-(259-I*38,25),A5%,PSET
510 NEXT I
520 TX=38:TY=22:AT=0:IF MEN>6THEN UX=1:UY=6 ELSE UX=20:UY=14
530 SB=525:PUT@A(525,16)-(623,38),B1,PSET:LOCATEUX,UY:COLOR2:PRINT"♠":LOCATETX,T
Y:COLOR3:PRINT"♥":COLOR7:LOCATE 0,0:PRINT"  ｼﾊﾞﾗｸ ｵﾏﾁｸﾀﾞｻｲ. ﾌﾞｻﾞｰｶﾞﾅｯﾀﾗ ｽﾀｰﾄ":CO
LOR3:IF S1=0 THEN LOCATE37,23:PRINT"&&"
540 IF S1=1 THEN LOCATE 38,23:PRINT"&"
550 '
560 '    **  ｶｸｽ ﾊﾞｼｮ   **
570 '
580 F1=2:F2=1:B1=B:B2=0:A2=A:E(1)=INT(RND(3)*MAL)+1
590 FOR I1=1 TO 5:FOR J=F1 TO B1+B2
600 E(J)=INT(RND(12)*MAL)+1
610 FOR L=1 TO J-1:IF E(J)=E(L) THEN  GOTO 600:ELSE NEXT L:NEXT J
620 FOR I=F2 TO B1+B2:G(E(I))=A2:NEXT I:B1=B1+B2:B2=INT((MAL-B)/4):F1=B1+1:F2=F1
630 A2=(A2+1)MOD6:IF A2=0 THEN 630 ELSE NEXT I1
640 IF MAL=15 AND B=3 THEN680
650 B3=1:FOR I2=F1 TO MAL:FOR I3=B3 TO MAL:E(I2)=I3:FOR I4=1 TO F1-1:IF E(I2)=E(
I4) THEN 660 ELSE NEXT I4:GOTO 670
660 NEXT I3
670 G(E(I2))=6:B3=I3+1:NEXT I2
680 FOR I=1 TO 100:BEEP 1:NEXT I:BEEP0
690 COLOR7:LOCATE 0,0:PRINT USING"##";MEN;:PRINT"        POINT      0      TOTAL
 ";:IF P#<0 THEN COLOR2
700 PRINT USING"######";P#:TX=38:TY=22:UU=0:SB2=0:PA=0:INTERVAL 1
710 FOR J=1 TO MAL
720 INTERVAL ON
730 TS=1:COLOR3:LOCATE TX,TY:PRINT"♥":TX1=TX:TY1=TY
740 TS=2:B$=INKEY$
750 TS=3:IF B$="" THEN 740
760 TS=4:IF B$="4" THEN TX=TX-1
770 TS=5:IF B$="6" THEN TX=TX+1
780 TS=6:IF B$="8" THEN TY=TY-1
790 TS=7:IF B$="2" THEN TY=TY+1
800 ON INTERVAL GOSUB1680
810 TS=8:LOCATE TX1,TY1:PRINT" "
820 TS=9:T1=SCREEN(TX,TY):IF T1=236 THEN 850
830 TS=10:IF T1=135 OR T1=232 OR T1=38 THEN TX=TX1:TY=TY1
840 TS=11:IF T1=64 THEN GOSUB 2050 ELSE GOTO730
850 INTERVAL STOP:LOCATE TX,TY:PRINT"♥":TXX=TX*16:TYY=TY*8
860 FOR I=1 TO 15
870 IF C(E(I))=TXX AND D(E(I))=TYY THEN UUX=UX-C(E(I))/16:UUY=UY-D(E(I))/8:GOTO
890
880 NEXT I
890 IF G(E(I))=1 THEN GOSUB 1360:PUT@A(C(E(I))-15,D(E(I))-8)-(C(E(I))-15+32,D(E(
I))-8+15),A1,PSET:GOTO950
900 IF G(E(I))=2 THEN GOSUB 1360:PUT@A(C(E(I))-15,D(E(I))-5)-(C(E(I))-15+29,D(E(
I))-5+10),A2,PSET:GOTO950
910 IF G(E(I))=3 THEN GOSUB 1410:PUT@A(C(E(I))-11,D(E(I))-6)-(C(E(I))-11+35,D(E(
I))-6+12),A3,PSET:GOTO950
920 IF G(E(I))=4 THEN GOSUB 1480:PUT@A(C(E(I))-13,D(E(I))-4)-(C(E(I))-13+38,D(E(
I))-4+16),A4,PSET:GOTO950
930 IF G(E(I))=5 THEN GOSUB 1360:PUT@A(C(E(I))-14,D(E(I))-5)-(C(E(I))-14+27,D(E(
I))-5+9),A5,PSET:GOTO950
940 IF G(E(I))=6 THEN GOSUB 1580:PUT@A(C(E(I))-27,D(E(I))-8)-(C(E(I))-27+51,D(E(
I))-8+15),A6,PSET:GOSUB 1340
950 SB1=0:IF G(E(I))<>A AND G(E(I))=6 THEN SB1=2:GOSUB1300
```

```
960 IF G(E(I))<>A AND G(E(I))<>6 THEN SB1=1:GOSUB1300
970 SB2=SB2+SB1:IF SB2<8 THEN P1=30:COLOR7
980 IF G(E(I))=A THEN P1=500:PA=PA+1:COLOR7:FOR BE=1 TO 10:BEEP1:NEXT BE:BEEP0
990 IF G(E(I))=6 AND SB2<8 THEN P1=-200:COLOR2
1000 IF G(E(I))<>A AND SB2>=8 THEN P1=(7-SB2)*200:COLOR2
1010 LINE(288,0)-(352,8),PSET,0,BF:LOCATE 16,0:PRINT USING"######";P1:IF SBL=1 T
HEN GOTO1060
1020 COLOR7:P#=P#+P1:IF P#<0 THEN COLOR2
1030 LOCATE 34,0:PRINT USING"######";P#
1040 IF PA=B THEN MEN=MEN+1:GOTO 1080
1050 NEXT J
1060 FOR JJ1=1 TO 300:NEXT JJ1:S1=S1+1:IF S1<3 THEN GOTO 400
1070 GOTO1090
1080 COLOR3:LOCATE10,0:PRINT"BONUS   3000":FOR II=1 TO 50 :NEXT II:COLOR7:LOCATE
 34,0:P#=P#+3000:PRINT USING "######";P#:FOR JJ1= 1 TO 500:NEXT JJ1:IF S1<3 THEN
 COLOR0:LOCATE37+S1,23:PRINT"&":GOTO 400
1090 FOR I2=1 TO 300:NEXT I2:CLS:LOCATE10,5:COLOR7:PRINT"ｱﾅﾀ ﾉ ﾄｸﾃﾝﾊ  ";:COLOR5:
IF P#<0 THEN COLOR2
1100 PRINT USING"######";P#;:COLOR7:PRINT" ﾃﾝ"
1110 IF P#<=0 THEN GOSUB 2100
1120 IF P#<=5000 THEN GOSUB 2110
1130 IF P#<=10000 THEN GOSUB 2160
1140 IF P#<=30000 THEN GOSUB 2200
1150 IF P#>30000 THEN GOSUB 2270
1160 '   **  ｷｬﾗｸﾀｰ ﾉ ﾀﾒ ﾉ ｻﾌﾞﾙｰﾁﾝ  **
1170 COLOR Z2:FOR I=1 TO Z1:READ AX,AY,BX,BY:LINE(AX+320,AY+100)-(BX+320,BY+100)
,PSET:NEXT I:RETURN
1210 '   **  ｹﾞｰﾑ ﾉ ｾﾂﾒｲ  **
1220 CLS:LOCATE 4,1:COLOR3:PRINT"♥";:COLOR7:PRINT" ･････ ｱﾅﾀｶﾞ ｳｺﾞｶｼﾏｽ":LOCATE 4
,3:COLOR2:PRINT"♠";:COLOR7:PRINT" ･････";:COLOR3:PRINT" ♥";:COLOR7:PRINT" ｦ ｵｯﾃｷ
ﾏｽ"
1230 LOCATE 4,5:COLOR5:PRINT"●";:COLOR7:PRINT" ･････ ｸﾀﾞﾓﾉｶﾞ ｶｸｻﾚﾃｲﾏｽ":LOCATE3,9
:COLOR3:PRINT"♥";:COLOR7:PRINT"ﾉ ｲﾄﾞｳ":COLOR4:LOCATE6,11:PRINT"8":LOCATE2,14:PRI
NT"4":LOCATE10,14:PRINT"6":LOCATE6,17:PRINT"2"
1240 Z1=10:Z2=7:GOSUB1170:LOCATE14,9:COLOR7:PRINT"1  ﾊｼﾞﾒﾆﾃﾞﾀ ｸﾀﾞﾓﾉﾉ ｼｭﾙｲ":LOCAT
E16,11:PRINT"ﾄｺｽｳｶﾞ ｱﾅﾀﾉｻｶﾞｽﾓﾉﾃﾞｽ｡":LOCATE14,13:PRINT"2  ";:COLOR2:PRINT"♠";:COL
OR7:PRINT"ﾆ ﾂｶﾏﾗﾅｲﾃﾞﾈ!"
1250 LOCATE14,15:PRINT"3  ";:COLOR1:PRINT"♠";:COLOR7:PRINT"ﾄ ｸﾀﾞﾓﾉﾃﾞﾀﾄｺﾛﾊ ﾄｵﾚ":
LOCATE16,17:PRINT"ﾏｾﾝ｡"
1260 LOCATE2,19:PRINT"4  ｲﾗﾅｲﾓﾉｦｻｶﾞｽﾄ ﾍﾋﾞｶﾞ ｳｺﾞｷﾏｽ｡":LOCATE2,21:PRINT"5  ﾍﾋﾞｶﾞ ﾋ
ﾀﾞﾘﾏﾃﾞｸﾙﾄ ｵﾜﾘﾃﾞｽ｡":LOCATE8,23:COLOR6:PRINT"ﾜｶｯﾀﾗ ﾅﾆｶｵｼﾃﾈ!!"
1270 IF INKEY$="" THEN 1270
1280 RETURN400
1290 '   **  ﾍﾋﾞ ﾉ ｲﾄﾞｳ  **
1300 FOR SBK=1 TO SB1:LINE (SB,16)-(SB+98,38),PSET,0,BF:PUT@A(SB-38,16)-(SB+98-3
8,38),B1%,PSET
1310 IF SB<80 THEN SBL=1:P1=-1200:COLOR2:RETURN 1010
1320 SB=SB-38:NEXT SBK:FOR SBK=1 TO 50:NEXT SBK:RETURN
1330 '   **  ﾓｸﾞﾗ ﾉ ﾜﾗｲｺﾞｴ  **
1340 FOR SBI=1 TO 20:BEEP1:FOR SBJ=1 TO4:NEXT SBJ:BEEP0:FOR SBJ=1 TO 2:NEXT SBJ:
NEXT SBI:RETURN
1350 '   **  ﾜｸ  **
1360 LINE(TX-1,TY-1)-(TX,TY),"■",0,B:TX=TX-2:TY=TY-2:LOCATE TX,TY:IF ABS(UUX)>2
AND ABS(UUY)>2 THEN RETURN
1370 IF UUX=-1 AND UUY=-1 THEN UX=UX-1:RETURN
1380 IF UUX=-1 AND UUY=0 THEN UX=UX-1:RETURN
1390 IF UUX=0 AND UUY=-1 THEN UX=UX+1
1400 RETURN
1410 LINE(TX-1,TY-1)-(TX+1,TY),"■",0,B:TX=TX-2:TY=TY-2:LOCATE TX,TY:IF ABS(UUX)>
2 AND ABS(UUY)>2 THEN RETURN
1420 IF UUX=-1 AND UUY=0 THEN UX=UX-1:RETURN
1430 IF UUX=-1 AND UUY=-1 THEN UX=UX-1:RETURN
1440 IF UUX=0 AND UUY=-1 THEN UY=UY-1:RETURN
1450 IF UUX=1 AND UUY=-1 THEN UX=UX+1:RETURN
1460 IF UUX=1 AND UUY=0 THEN UX=UX+1
1470 RETURN
1480 LINE(TX-1,TY-1)-(TX+1,TY+1),"■",0,B:TX=TX-2:TY=TY-2:LOCATE TX,TY:IF ABS(UUX
```

```
))2 AND ABS(UUY))2 THEN RETURN
1490 IF UUX=-1 AND UUY=1 THEN UX=UX-1:RETURN
1500 IF UUX=-1 AND UUY=0 THEN UX=UX-1:RETURN
1510 IF UUX=-1 AND UUY=-1 THEN UX=UX-1:RETURN
1520 IF UUX=0 AND UUY=1 THEN UY=UY+1:RETURN
1530 IF UUX=0 AND UUY=-1 THEN UY=UY-1:RETURN
1540 IF UUX=1 AND UUY=1 THEN UX=UX+1:RETURN
1550 IF UUX=1 AND UUY=0 THEN UX=UX+1:RETURN
1560 IF UUX=1 AND UUY=-1 THEN UX=UX+1
1570 RETURN
1580 LINE(TX-2,TY-1)-(TX+1,TY),"■",0,B:TX=TX-3:TY=TY-2:LOCATE TX,TY:IF ABS(UUX)>
3 AND ABS(UUY)>2 THEN RETURN
1590 IF UUX=-2 AND UUY=0 THEN UX=UX-1:RETURN
1600 IF UUX=-2 AND UUY=-1 THEN UX=UX-1:RETURN
1610 IF UUX=-1 AND UUY=0 THEN UY=UY+1:RETURN
1620 IF UUX=-1 AND UUY=-1 THEN UY=UY-1:RETURN
1630 IF UUX=0 AND UUY=-1 THEN UY=UY-1:RETURN
1640 IF UUX=1 AND UUY=0 THEN UX=UX+1:RETURN
1650 IF UUX=1 AND UUY=-1 THEN UX=UX+1
1660 RETURN
1670 ´   **  ♠ ﾉ ｲﾄﾞｳ  **
1680 UX1=UX:UY1=UY:IF MEN<4 OR MEN>6 AND MEN<10 THEN AT=(AT+1)MOD15:IF AT=14 THE
N GOSUB 2030
1690 IF UX<TX AND UY<TY THEN UX=UX+1:UY=UY+1:GOTO 1770
1700 IF UX<TX AND UY>TY THEN UX=UX+1:UY=UY-1:GOTO 1770
1710 IF UX<TX AND UY=TY THEN UX=UX+1:GOTO 1770
1720 IF UX>TX AND UY<TY THEN UX=UX-1:UY=UY+1:GOTO1770
1730 IF UX>TX AND UY>TY THEN UX=UX-1:UY=UY-1:GOTO 1770
1740 IF UX>TX AND UY=TY THEN UX=UX-1:GOTO 1770
1750 IF UX=TX AND UY<TY THEN UY=UY+1:GOTO 1770
1760 IF UX=TX AND UY>TY THEN UY=UY-1:GOTO 1770
1770 U1=SCREEN(UX,UY)
1780 IF U1=233THEN UU=1
1790 IF U1=135 OR U1=236 OR U1=38 THEN GOSUB 1940
1800 COLOR1:LOCATE UX1,UY1:PRINT"♠"
1810 COLOR2:LOCATE UX,UY:PRINT"♠":COLOR3
1820 IF UU=1 THEN COLOR1:LOCATE UX,UY:PRINT"♠":COLOR2:LOCATE TX,TY:PRINT"♠":INTE
RVAL OFF:IF S1=3 THEN RETURN 1090ELSE RETURN 1060
1830 IF TS=1 THEN RETURN 730
1840 IF TS=2 THEN RETURN 740
1850 IF TS=3 THEN RETURN 750
1860 IF TS=4 THEN RETURN 760
1870 IF TS=5 THEN RETURN 770
1880 IF TS=6 THEN RETURN 780
1890 IF TS=7 THEN RETURN 790
1900 IF TS=8 THEN RETURN 810
1910 IF TS=9 THEN RETURN 820
1920 IF TS=10 THEN RETURN 830
1930 IF TS=11 THEN RETURN 840
1940 IF UX1>TX THEN UX=UX1-1
1950 IF UX1<TX THEN UX=UX1+1
1960 U2=SCREEN(UX,UY1):IF U2=236 OR U2=135 OR U2=38 THEN 1980
1970 UY=UY1:RETURN
1980 IF UY1>TY THEN UY=UY1-1
1990 IF UY1<TY THEN UY=UY1+1
2000 UX=UX1:U2=SCREEN(UX,UY):IF U2=236 OR U2=135 OR U2=38 THEN UY=UY1
2010 RETURN
2020 ´   **  @ ﾉ ﾋｮｳｼﾞ  **
2030 ATX=INT(RND(B)*38)+1:ATY=INT(RND(B)*18)+6:ATS=SCREEN(ATX,ATY):IF ATS=38 OR
ATS=135 OR ATS=232 OR ATS=233 OR ATS=236 THEN RETURN ELSE COLOR7:LOCATE ATX,ATY:
PRINT"@":RETURN
2040 ´   **  ♥ ﾉ ﾜｰﾌﾟ  **
2050 INTERVAL STOP:COLOR3:LOCATETX,TY:PRINT"♥":FOR HAT=1 TO 50:NEXT HAT:LOCATE T
X,TY:PRINT" "
2060 TX=INT(RND(A)*38)+1:TY=INT(RND(A)*18)+6:HAS=SCREEN(TX,TY):IF HAS=135 OR HAS
=236 OR HAS=64 OR HAS=38 OR HAS=232 THEN GOTO 2060
2070 COLOR3
```

```
2080 LOCATE TX,TY:PRINT"♥": RETURN720
2090 '    **  サイコ" ノ エ    **
2100 FOR TN=1 TO 100:NEXT TN:RESTORE 2810:COLOR3:FOR TT=1 TO 62:READ TX,TY:LOCAT
E TX,TY:PRINT"♥":NEXT TT:GOTO2310
2110 FOR TN=1 TO 100:NEXT TN:COLOR5:LOCATE33,13:PRINT"●"
2120 FOR TT=0 TO 22
2130 FOR TN=1 TO 10:NEXT TN:COLOR3:LOCATETT+2,13:PRINT" ♥":FOR TN=1 TO 10:NEXT T
N:LOCATETT+1,13:COLOR1:PRINT"♠";:COLOR2:PRINT"♠"
2140 NEXT TT
2150 FOR TN=1 TO 100:NEXT TN:LOCATE25,13:PRINT" ":SYMBOL(432,103),"♥",1,2,3,3,PS
ET:FOR TN=1 TO 100:NEXT TN:COLOR7:LOCATE12,15:PRINT"コンシ"ョウ カ" タリナイ !!":GOTO2310
2160 FOR TN=1 TO 100:NEXT TN:COLOR5:LOCATE33,13:PRINT"●"
2170 FOR TT=1 TO 29
2180 FOR TN=1 TO 10:NEXT TN:COLOR3:LOCATETT+2,13:PRINT" ♥":FOR TN=1 TO 10:NEXT T
N:LOCATETT+1,13:COLOR1:PRINT"♠";:COLOR2:PRINT"♠":NEXT TT
2190 FOR TN=1 TO 10:NEXT TN:LOCATE32,13:COLOR3:PRINT" ♥":PUT@A(517,100)-(568,115
),A6,PSET:GOSUB1340:FOR TN=1 TO 100:NEXT TN:COLOR7:LOCATE12,15:PRINT"モウチョット カ"ンハ
"ッテ":GOTO2310
2200 FOR TN=1 TO 100:NEXT TN:LOCATE33,13:COLOR5:PRINT"●":FOR TT=0 TO 29
2210 FOR TN=1 TO 10:NEXT TN:COLOR3:LOCATETT+2,13:PRINT" ♥":FOR TN=1 TO 10:NEXT T
N:LOCATETT+1,13:COLOR1:PRINT"♠";:COLOR2:PRINT"♠":NEXT TT
2220 FOR TN=1 TO 10:NEXT TN:LOCATE32,13:COLOR3:PRINT" ♥":PUT@A(525,99)-(560,111)
,A3,PSET:LOCATE35,13:PRINT"♥":FOR TM=1 TO 40:NEXT TM:COLOR2:FOR TN=0 TO 2:LOCATE
30-TN,13:PRINT"♠ ":FOR TM=1 TO 10:NEXT TM:NEXT TN
2230 FOR TN=1 TO 50:NEXT TN:COLOR3:LOCATE35,13:PRINT"  ♥":FOR TN=1 TO 70:NEXT TN
:LOCATE35,13:PRINT"♥  "
2240 COLOR2:XT=27:FOR TN=1 TO 9:LINE(561-TN*16*3,99)-(609-TN*16*3,111),PSET,0,BF
2250 PUT@A(525-TN*16*3,99)-(560-TN*16*3,111),A3,PSET:FOR TL=1 TO 10:NEXT TL:FOR
TM=0 TO 2:LOCATEXT,13:PRINT"♠ ":XT=XT-1:FOR TP=1 TO 10:NEXT TP:NEXT TM:NEXT TN
2260 LINE(81,99)-(129,111),PSET,0,BF:PUT@A(45,99)-(80,111),A3,PSET:FOR TT=1 TO 1
00:NEXT TT:COLOR7:LOCATE15,15:PRINT"ヤルモンタ"ネ!":GOTO2310
2270 FOR TN=1 TO 100:NEXT TN:LOCATE33,13:COLOR5:PRINT"●":FOR TT=1 TO 29
2280 FOR TN=1 TO 10:NEXT TN:COLOR3:LOCATETT+2,13:PRINT" ♥":FOR TN=1 TO 10:NEXT T
N:LOCATETT+1,13:COLOR1:PRINT"♠";:COLOR2:PRINT"♠":NEXT TT:FOR TN=1 TO 10:NEXT TN:
LOCATE32,13:COLOR3:PRINT" ♥"
2290 LOCATE33,13:PRINT" ":SYMBOL(520,96),"♥",4,2,3,0:FOR TN=1 TO 50:NEXT TN:SYMB
OL(520,96),"♥",4,2,0,0:SYMBOL(520,81),"♥",4,2,3,0:FOR TN=1 TO 50:NEXT TN:SYMBOL(
520,81),"♥",4,2,0,0:SYMBOL(520,96),"♥",4,2,3,0
2300 LOCATE31,13:PRINT" ":SYMBOL(496,111),"♠",1,2,2,1:FOR TL=0 TO 28:LOCATE30-TL
,13:PRINT" ":SYMBOL(480-TL*16,111),"♠",1,2,5,1:NEXT TL:FOR TN=1 TO 100:NEXT TN:C
OLOR7:LOCATE12,15:PRINT"アナタッタラ サイコーヨ!":GOTO2310
2310 '    **  サイコ" ノ メッセーシ"   **
2320 LOCATE 7,20:COLOR7:PRINT"モウイチト" ヤッテミマスカ?  YES ナラ <Y>":LOCATE7,22:PRINT"NO ナ
ラ <N> ヲ オシテクタ"サイ。";
2330 Z1$=INKEY$:IF Z1$="" THEN 2330
2340 IF Z1$="Y" OR Z1$="y" THEN PRINT"Y":S1=0:FOR ZI=1 TO 200:NEXT ZI:MEN=1:P#=0
:RETURN 400
2350 IF Z1$="N" OR Z1$="n" THEN PRINT"N":FOR ZI=1 TO 200:NEXT ZI:GOTO2370
2360 GOTO2330
2370 CLS:SYMBOL(130,80),"オ シ マ イ",7,6,4,0
2380 IF INKEY$="" THEN 2380ELSE END
2390 '    **  ヘビ"  **
2400 DATA -160,-40,-157,-42,-157,-42,-152,-43,-152,-43,-146,-43,-146,-43,-142,-4
1,-142,-41,-139,-38
2410 DATA -138,-38,-136,-34,-136,-34,-133,-32,-133,-32,-129,-30,-129,-30,-124,-3
0,-124,-30,-121,-31
2420 DATA -121,-31,-118,-33,-118,-33,-115,-35,-115,-35,-111,-38,-111,-38,-105,-4
0,-105,-40,-99,-40
2430 DATA -99,-40,-95,-38,-95,-38,-93,-36,-93,-36,-91,-34,-91,-34,-89,-31,-89,-3
1,-87,-30
2440 DATA -87,-30,-84,-30,-84,-30,-81,-31,-81,-31,-78,-33,-78,-33,-75,-35,-75,-3
5,-75,-33
2450 DATA -75,-33,-76,-30,-76,-30,-78,-28,-78,-28,-81,-27,-81,-27,-86,-26,-86,-2
6,-90,-26
2460 DATA -91,-26,-94,-27,-94,-27,-96,-29,-96,-29,-99,-31,-99,-31,-103,-31,-103,
-31,-106,-30
2470 DATA -106,-30,-110,-28,-110,-28,-116,-26,-116,-26,-123,-25,-123,-25,-129,-2
```

```
5,-129,-25,-134,-26
2480 DATA -134,-26,-138,-27,-138,-27,-143,-30,-143,-30,-146,-32,-146,-32,-150,-3
3,-150,-33,-154,-34
2490 DATA -154,-34,-158,-35,-158,-35,-159,-36,-159,-36,-152,-38,-152,-38,-159,-4
0,-150,-40,-148,-40
2500 DATA -156,-46,-135,-40,-149,-45,-143,-47,-143,-47,-134,-45,-134,-45,-141,-4
2,-153,-38,-157,-38
2510 DATA -157,-38,-160,-38,-160,-38,-162,-37,-162,-37,-166,-38,-166,-38,-169,-3
9,-168,-39,-171,-39
2520 DATA -138,-31,-136,-31,-132,-28,-130,-28,-120,-27,-118,-27,-114,-32,-112,-3
2,-106,-37,-104,-37
2530 DATA -99,-34,-97,-34,-92,-29,-90,-29,-84,-29,-83,-29
2540 DATA -99,-34,-97,-34,-92,-29,-90,-29,-84,-29,-83,-29
2550 DATA  83,-25, 83,-30, 83,-30, 85,-33, 85,-33, 89,-35, 89,-35, 94,-36, 94,-3
6, 98,-36
2560 DATA  98,-36, 103,-35, 103,-35, 107,-33, 107,-33, 110,-31, 110,-31, 110,-25
, 110,-25, 83,-25
2570 DATA  110,-25, 110,-21, 110,-21, 106,-23, 106,-23, 104,-21, 104,-21, 100,-2
3, 100,-23, 98,-21
2580 DATA  98,-21, 98,-25, 94,-25, 94,-21, 94,-21, 91,-23, 91,-23, 88,-21, 88,-2
1, 85,-23
2590 DATA  85,-23, 83,-21, 83,-21, 83,-25,108,-31,112,-31,85,-31,81,-31, 91,-31,
 93,-31, 99,-31, 101,-31
2600 DATA  91,-27, 91,-25, 91,-25, 101,-25, 101,-25, 101,-27, 101,-27, 92,-27, 9
6,-27, 96,-25
2610 DATA  106,-27, 121,-29, 107,-26, 121,-25, 85,-27, 71,-29, 84,-26, 71,-25
2620 '   ** ﾌﾞﾄﾞｳ  **
2630 DATA 108,120,0,1,111,117,.43,.23,112,122,.72,.5,117,120,.7,.25,117,117,.63,
.15,105,124,0,.75
2640 DATA -198,15,-193,13,-193,13,-198,11,-189,16,-195,14
2650 '   ** ﾐｶﾝ  **
2660 DATA -107, 23,-107, 23,-97, 22,-97, 22,-93, 24,-93, 24,-109, 21,-109, 21,-1
01, 25,-101, 25
2670 '   ** ﾒﾛﾝ  **
2680 DATA 103,15,103,13,97,13,108,13
2690 DATA  91, 23, 99, 19, 97, 17, 102, 22, 98, 23, 106, 21, 104, 18, 111, 22, 1
08, 24, 116, 21
2700 DATA  97, 27, 100, 24, 99, 25, 105, 27, 88, 22, 95, 25, 103, 26, 111, 25
2710 '    ** ｸﾘ  **
2720 DATA  182, 25, 183, 23, 183, 23, 186, 22, 186, 22, 189, 21, 189, 21, 193, 2
0, 193, 20, 195, 19
2730 DATA195,19,197,20,197,20,201,21,201,21,204,22,204,22,207,23,207,23,208,25
2740 '   ** ﾔｼﾞﾙｼ  **
2750 DATA -266, 15,-167, 15,-216,-2,-216, 31,-225,3,-216,-2,-216,-2,-207,3,-176,
12,-167,15,-167,15,-176,18,-216,31,-207,26,-216,31,-225,26,-257,18,-266,15,-266,
15,-257,12
2760 '   ** ｶｸｽ ﾄｺﾛ  **
2770 DATA 112,168,208,144,320,160,448,136,544,96,352,104,240,96,128,120,144,64,4
00,72,528,168,96,88,512,56,288,64
2780 DATA528,112,240,160,352,168,480,160,64,136,160,104,320,120,416,136,448,96,3
68,64,256,80,80,88,48,168,176,64,560,168
2790 DATA 80,176,208,160,320,96,128,112,240,120,320,136,400,168,432,112,512,136,
496,80,208,80,336,56,112,64,544,160,544,56,48,88
2800 '    **  ｻｲｺﾞ  **
2810 DATA 2,12,3,11,4,10,5,10,6,10,6,11,6,12,5,13,4,14,3,15,2,16,4,12,8,10,9,11,
9,9,10,10,16,10,16,11,15,12,15,13,14,14,13,15,12,16,13,12,14,13,15,14,16,15
2820 DATA 19,12,20,12,21,12,22,12,23,12,21,10,21,11,21,13,21,14,21,15,20,16,27,1
2,28,12,29,12,30,12,26,10,26,11,26,12,26,13,26,14,26,15,27,16,28,16,29,16,30,16,
33,10,33,11,33,12,33,13,33,14,33,15,33,16,34,13,35,14,36,15
3010 '    **************************
3020 '    **                      **
3030 '    **    ｻｶﾞｼﾓﾉ  ｹﾞｰﾑ     **
3040 '    **                      **
3045 '    **         by ﾌｻｺ ﾋﾗｶﾞ  **
3048 '    **                      **
3050 '    **************************
6060 '(c)KANTO DENSHI CORPORATION 1983
```

川下りゲーム

くすだ　あきら

舟（◆）で川を下りながら，障害物（■）にぶつからないようにして宝物（＄）を拾うというゲームだ。腕に自信のある人はレベル2，あまり腕に自信のない人はレベル1を選ぶ。レベル1は ← ↓ → のキーで，レベル2は ← → だけで舟をあやつる。舟の方向は一度指定すると，次のキーが押されるまでその方向を維持する（これを慣性航行するという）。

さて，ゲームの神髄は川を下ることにある。宝物を拾うことにあまり熱中すると，岩や川岸に当たって舟は沈没してしまう。沈没すると，せっかく取った宝物は全部川の中に落ちてしまう。がんばって画面をクリアしてください。クリアすると，川幅はだんだん狭くなり，岩の数も増えてくる。しかし，ただ難しくなるだけでなく，2画面クリアするごとに，楽しいボーナスチャレンジがある。おまけとして，10,000点以上になると舟が一隻ふえる。

1プレイ終えて，スコアが2,000点以上なら，名前を登録できる(ベスト5まで)。

プログラムの構成を以下に示す。

100：初期
110〜150：初期タイトル画面
160〜240：ゲームの説明
250〜300：レベル入力
320：ゲームの初期
340〜380：ゲームの初期画面
390〜480：ゲームのメインルーチン
490：ゴール表標サブ
510〜530：レベル1処理
540〜560：レベル2処理
580〜630：画面クリア
650〜660：岩とぶつかる
680〜800：ボーナスチャレンジ
820〜840：とっても腕の良い人のためのルーチン
860〜：END処理

なお，ゲームをやめるときは，STOP キーを押す。

```
100 DEFINT A-Z:DEFDBL T,H:COLOR7,0:WIDTH40,25:CONSOLE0,25,0:DIM HT(6),HT$(6):FOR
 I=0 TO 6:HT(I)=2000:HT$(I)="":NEXT
110 CLS:LINE(0,3)-(39,15),"$",6,BF:LINE(3,6)-(36,11)," ",,BF
120 COLOR5:SYMBOL(80,60),"ｶﾜｸﾀﾞﾘ ｹﾞｰﾑ",5,3
130 LOCATE7,18:COLOR6:PRINT"ｾﾂﾒｲ ｲﾘﾏｽｶ ?":LOCATE7,20:PRINT"(ｲﾙ=Y,ｲﾗﾅｲ=N ｦ ｵｼﾃｸﾀﾞ
ｻｲ)";CHR$(7)
140 Z$=INPUT$(1):LOCATE20,18:IF " ">Z$ THEN150 ELSE PRINT Z$;:IF Z$="Y" OR Z$="y
" THEN FOR I=0 TO500:NEXT:GOTO160 ELSE IF Z$="N" OR Z$="n" THEN FOR I=0 TO500:NE
XT:GOTO250
150 BEEP:GOTO140
160 CLS:PRINT:PRINT" ● ｺﾉｹﾞｰﾑﾊ (  ) ﾉﾌﾈﾃﾞ ｶﾜｦｸﾀﾞﾙｹﾞｰﾑﾃﾞｽ":LOCATE12,1:COLOR7:PRINT
"◆":COLOR6:LOCATE0,2
170 PRINT:PRINT" ● ｶﾜｷﾞｼ ﾏﾀﾊ ｼｮｳｶﾞｲﾌﾞﾂ (  ) ﾆﾌﾞﾂｶﾙﾄ ﾌﾈﾊ":PRINT"    ｼｽﾞﾐﾏｽ  (ﾀﾀﾞｼ ﾎ
ﾞｰﾅｽﾁｬﾚﾝｼﾞﾉﾄｷﾊ ﾉｿﾞｸ)":LOCATE24,3:COLOR1,7:PRINT"■":COLOR6:LOCATE0,5
180 PRINT:PRINT" ● ($) ｦﾋﾛｳﾄ ﾎﾞｰﾅｽﾃﾝﾄｼﾃ10ﾃﾝ-90ﾃﾝ ﾉﾄﾞﾚｶ":PRINT"    ﾓﾗｴﾏｽ"
190 PRINT:PRINT" ● ﾌﾈﾊ ｶﾝｾｲ ｺｳｺｳ ｼﾏｽ"
200 PRINT:PRINT" ● ｹﾞｰﾑﾆﾊ ﾚﾍﾞﾙ(1,2)ｶﾞｱﾘﾏｽ":PRINT"    ﾚﾍﾞﾙ1:ﾐｷﾞ,ｼﾀ,ﾋﾀﾞﾘ,ﾆｲｹﾏｽ":PRI
NT"    ﾚﾍﾞﾙ2:ﾐｷﾞ,ﾋﾀﾞﾘ,ﾀﾞｹﾃﾞｽ"
210 PRINT:PRINT" ● ﾌﾈｦｳｺﾞｶｽ ｺﾏﾝﾄﾞﾊ ｶｰｿﾙｲﾄﾞｳ ﾉ ｷｰﾃﾞｽ"
220 LOCATE6,20:PRINT SPC(26):LOCATE6,20:COLOR13:PRINT" ﾖｹﾚﾊﾞ ｽﾍﾟｰｽｷｰ ｦ ｵｼﾃｸﾀﾞｻｲ
":FOR I=1 TO 800:NEXTI:COLOR6
230 Z$=INKEY$:IFZ$=" " THEN GOTO250
240 GOTO220
250 COLOR7,0:CLS:IF TIME>30000 THEN TIME$="00:00:00"
260 LOCATE5,18:COLOR7:PRINT"ﾚﾍﾞﾙ (1 ﾏﾀﾊ2) ｦｲﾚﾃｸﾀﾞｻｲ"
270 LOCATE6,21:PRINT"ﾚﾍﾞﾙ ?";CHR$(7)
280 Z$=INPUT$(1):IF " ">Z$ THEN300 ELSE LOCATE13,21:PRINT Z$
290 IFZ$="1" OR Z$="2" THEN L=VAL(Z$):FOR I=0 TO500:NEXT:GOTO320
300 BEEP:GOTO280
310 '-------------------------------------------------
320 T3=-13:H=17:XX=20:X=11:M=1:C=XX:K=313:T=0:T2=0:T1=0:N=4:B=2:XP=11:AAA=1:T4=0
:TH=10000
330 '-------------------------------------------------
340 CLS:A$="":AA$=""
350 LINE(0,0)-(39,23),"■",1,B:LINE(X+2,23)-(X+H-1,24)," ",,B
360 LOCATE20,2:COLOR5:PRINT"ｱﾅﾀﾉﾌﾈﾉ ﾉｺﾘ";B:LOCATE5,4:PRINT"ｽﾀｰﾄ!",M;"ﾒﾝ":LOCATEX
X,10:COLOR7:PRINT"◆":LOCATE7,6:COLOR7:PRINT"ｱﾅﾀﾉﾃﾝ";T:LOCATE7,8:COLOR2:PRINT"ﾊｲｽ
ｺｱｰ";HT(1);"･･･";HT$(1)
370 Z$=INKEY$
380 IFZ$=CHR$(28) OR Z$=CHR$(29) THEN370 ELSEFOR I=0 TO1500:NEXTI
390 COLOR7:LOCATEXX,10:PRINT"◆":IFD=36 THENLOCATEXX-1,9:T2=INT(RND(TIME)*9+1)*10
:T4=T4+T2:PRINTT2
400 IFT1=K THEN580
410 ONL GOSUB510,540
```

```
420 D=SCREEN(C,11):IFD=135 THEN650
430 A=INT(RND(TIME)*3):IFA=0 THENX=X+1:IFX+H>37 THENX=37-H
440 IFA=1 THENX=X-1:IFX<4 THENX=4
450 COLOR1:T1=T1+1:IFT1=K-13 THEN490 ELSELOCATEX,25:PRINT"■■":LOCATEX+H,23:PRINT
"■■"
460 IFTIME MODN=0 THENLOCATEX+INT(RND(TIME)*(H-3)+3),23:PRINT"■"
470 IFTIME MOD7=0 THENLOCATEX+INT(RND(TIME)*(H-3)+3),23:COLOR6:PRINT"$"
480 LOCATEXX,9:PRINT" ":XX=C:T3=T3+1:GOTO390
490 LINE(4,24)-(X+1,24),"■":LINE(X+H,24)-(39,24),"■":LOCATE0,24:COLOR9:PRINT" ｺﾞ
ｰﾙ":COLOR1:GOTO460
500 '---------------------------------------------------
510 A$=INKEY$:IFA$="" THENA$=AA$
520 IFA$=CHR$(28) THENC=C+1 ELSEIFA$=CHR$(29) THENC=C-1
530 AA$=A$:RETURN
540 AA$=INKEY$:IFAA$=CHR$(28) ORAA$=CHR$(29) THENA$=AA$
550 IFA$=CHR$(28) THENC=C+1 ELSEIFA$=CHR$(29) THENC=C-1
560 RETURN
570 '---------------------------------------------------
580 FOR I=0 TO 1500:NEXTI:A$="":AA$="":D=0:T1=0:T3=T3+T4:H=H-2:IF H<6 THENH=7
590 M=M+1:IF M MOD 2=1THEN GOSUB680:T=T+T5
600 P=M:IF M>6 THEN P=6:N=N-1:IFN<1 THENGOTO820
610 T=T+T3:IFT>TH THENIFAAA=1 THEN AAA=0:B=B+1
620 T3=-13:C=20:XX=20:X=10+P:XP=X:D=0:T1=0:T4=0
630 GOTO340
640 '---------------------------------------------------
650 FORI=0 TO8:BEEP1:LOCATEXX,10:PRINT"+":FORJ=0 TO30:NEXTJ:BEEP0:LOCATEXX,10:PR
INT"X":FORJ=0 TO30:NEXTJ:NEXTI:CLS
660 GOTO860
670 '---------------------------------------------------
680 CLS:TT=0:T5=0:T6=0:LINE(0,5)-(39,15),"$",6,B:LINE(1,6)-(38,14),"$",,B
690 COLOR6:SYMBOL(50,80),"ﾎﾞｰﾅｽ ﾁｬﾚﾝｼﾞ!!",5,3
700 LOCATE9,18:COLOR5:PRINT"$ ｦﾄｯﾃ ｸﾀﾞｻｲ"
710 FOR I=0 TO 2000:NEXTI:CLS
720 XX=20:C=20
730 LOCATEXX,10:COLOR7:PRINT"◆":IFD=36 THENLOCATEXX-1,9:T6=INT(RND(TIME)*9+1)*10
:T5=T5+T6:PRINTT6
740 IFTT=K THEN800
750 ONL GOSUB510,540
760 D=SCREEN(C,11):IFD=135 THENGOTO800
770 TT=TT+1:COLOR1:LOCATEINT(RND(TIME)*39),25:PRINT"■"
780 COLOR6:LOCATEINT(RND(TIME)*39),23:PRINT"$"
790 LOCATEXX,9:PRINT" ":XX=C:GOTO730
800 FOR I=0 TO 1000:NEXTI:CLS:LINE(0,5)-(39,15),"$",6,B:LINE(1,6)-(38,14),"$",,B
:COLOR5:SYMBOL(50,80),"ﾎﾞｰﾅｽﾃﾝ",5,3:Z$=STR$(T5):SYMBOL(300,80),Z$,5,3:FOR I=0 TO
2000:NEXTI:RETURN
810 '---------------------------------------------------
820 CLS:IFL=2 THENSYMBOL(50,60),"ﾏ･･･ﾏｲｯﾀ････!!",5,3:FOR I=0 TO2000:NEXTI:CLS:T=
T+T3:GOTO870
830 L=2:SYMBOL(50,20),"ﾖｸﾃﾞｷﾏｼﾀ",5,3:SYMBOL(50,70),"ﾚﾍﾞﾙ 2 ﾆ ﾁｮｳｾﾝ",5,3:SYMBOL(5
0,95),"ｼﾏｽｶ?",5,3:FOR I=0 TO 2000:NEXTI:SYMBOL(50,150),"ｼﾏｽﾈ",5,3:FOR I=0 TO2000
:NEXTI:CLS:SYMBOL(50,60),"ﾃﾞﾜ････",5,3:SYMBOL(50,100),"ｶﾞﾝﾊﾞｯﾃ!!",5,3
840 FOR I=0 TO2000:NEXTI:CLS:GOTO320
850 '---------------------------------------------------
860 T=T+T3:B=B-1:IFB>-1 THENA$="":AA$="":D=0:X=XP:T1=0:T3=-13:C=20:XX=20:GOTO340
870 CLS:LINE(0,0)-(39,20),"@",7,B
880 LOCATE5,3:COLOR2:PRINT"ｽｺｱｰ":LOCATE20,3:PRINT"ﾀﾞｼﾀﾋﾄ "
890 FORI=1 TO5:LOCATE3,I*2+3:PRINTI;HT(I):LOCATE20,I*2+3:PRINTSPC(18):LOCATE20,I
*2+3:PRINTHT$(I):NEXTI
900 LOCATE5,15:COLOR5:PRINT"ｱﾅﾀﾉ ﾃﾝｽｳ ﾊ";T:LOCATE5,2:PRINTM;"ﾒﾝﾃﾞ ｵﾜﾘ"
910 I=0
920 I=I+1:IF T>HT(I) THEN J=6:GOTO930 ELSEIFI<5 THEN920 ELSEGOTO960
930 FORJ=5 TOI STEP-1:HT(J)=HT(J-1):HT$(J)=HT$(J-1):NEXTJ:GOTO940
940 HT(I)=T:LOCATE5,21:PRINT"ｱﾅﾀﾉﾅﾏｴｦ ｲﾚﾃ RETURN ｷｰｦ";:LOCATE6,23:LINE INPUT"ｵｼﾃ
ｸﾀﾞｻｲ ";HT$(I):HT$(I)=LEFT$(HT$(I),16)
950 COLOR2:FORI=1 TO5:LOCATE3,I*2+3:PRINTI;HT(I):LOCATE20,I*2+3:PRINTSPC(18):LOC
ATE20,I*2+3:PRINTHT$(I):NEXTI
960 LOCATE5,18:COLOR7:PRINT"ﾓｳｲﾁﾄﾞ ﾔﾘﾀｲﾋﾄﾊ ｽﾍﾟｰｽｷｰ ｦ ｵｾ"
```

```
970 IF INKEY$=" " THENCLS:GOTO260 ELSE970
980 END
1010 '---------------------------------
1020 '
1030 '      KAWAKUDARI.GAME
1040 '
1050 '         82/03/15
1060 '
1070 '      Written by K.K
1080 '(c)KANTO DENSHI CORPORATION 1983
1090 '---------------------------------
```

ソフト情報

ディスケットの病院 HOSPITAL

5インチディスケットの種々の問題（ソフト的なディスケットクラッシュなど）に対処できるプログラム"HOSPITAL"と，それに連動する6つのプログラムと，改造用の3つのプログラムが収められたFM-8，FM-7用のディスクユーティリティが，cosmos™ 岡山（〒700 岡山市南方5-6-5 今田ビル2F ☎0862(54)7474）から発売された。

プログラムの内容は，以下のとおり。

①受付……プログラムの内容説明

②健康診断……ディスケット上の機械的なキズ（Physical Error）を調べる。

③産婦人科……F-BASIC用のフォーマット，システムコピー，初期化とVolcopyなどを実行する。

④胃腸科……F-BASIC用ディスクのFAT/Directryを解読。

⑤X線診断……FM-8仕様に限らず，両面倍密（40T），片面倍密（35T）ディスケットの任意のセクタをCRTに表示。

⑥整形外科……⑤と同じく，任意のセクタをバイト単位でスクリーン上で修正できる。

⑦脳神経外科……各クラスタの最初のセクタを表示する。また，このセクタの前半分と直前のクラスタの最後の半セクタも表示可。プログラム収納の構造が調べられ，ツブれたプログラムの修復，他社仕様のディスケットの移植の準備に使われる。

⑧人工受精……片倍（35T）ディスケットの任意の面の全Copy。

⑨心臓移植……PC-8801（NEC）仕様のプログラムファイルをFM-8仕様にCopy。

⑩退院……終了。

なお，プログラムの大部分はF-BASICで書かれているので，多少の手直しで応用範囲を広げられる。定価は15,000円（ディク版）。

音楽も楽しめるコンピュータダイスプレーゲーム　ウェディングレース

チャンピオンソフト

「ウエディングレース」は，1～4人で楽しめる〝人生すごろくゲーム〟のコンピュータ版で，この世に生まれてからいかに早く結婚にゴールインするかを競うゲームだ。

フニクリフニクラの軽快なメロディーとともにゲームの幕があく。

ゲームのルール表示が終わると漢字表示による画面が表示され，いよいよゲームの準備にとりかかる。まずプレーヤーの人数（1～4人）を，次いでゲームをすすめる順番をコンピュータに登録する。

各プレーヤーは100万円の持ち点でスタートし，PLAY CARD（1～6のキーを押す）を引いてゲームをすすめていく。

生まれてからお宮参りに行くところからゲーム開始。結婚にゴールインする間には入学祝金をもらったり，マイコンプログラムコンテストに入賞するなど，臨時収入があるが，一方でステレオを買ったり海外旅行をしたり，思いがけない支出がかさむ。

宝くじに当たると2,000万円の臨時収入となるので，このチャンスは絶対逃さないこと！

ゲームの途中で持ち点がマイナスになると破産し，〝運命〟の曲とともに地獄行きとなり失格。運よくゴールインするとウエディングマーチが鳴り響き天国行きというわけ。1着50万円，2着30万円，3着10万円のボーナスがつき，ゴールインしたとき持ち点の多い順に勝敗が決まる。

ガールフレンドと，ボーイフレンドと，あるいは家族そろってワイワイ言いながらやるのも，結構おもしろい。

プログラムの構成は，

行300～480　ゲームルール説明
1000～1450　画面表示
2000～2320　プレーヤーの人数，ゲームの順番登録
3000～7200　ゲーム進行
12000　宝くじが当ったときのMUSIC
12500　宝くじがはずれたときのMUSIC
12600　破産したときのMUSIC（運命）
12700　結婚にゴールインしたときのMUSIC（ウエディングマーチ）
50000～50060　漢字表示ルーチン
60010～60395　漢字データ
61010～61850　ゲーム進行データ

＊　＊　＊

なお，この「ウエディングレース」は，エヌ・ティ・エル産業㈱や全国のマイコンショップで販売されている（定価3,500円）。

詳しくは下記へ。

エヌ・ティ・エル産業㈱
〒657 神戸市灘区篠原北1-5-13
☎(078)881-1914

```
1 REM  ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
2 REM  ◆                            ◆
3 REM  ◆          FM-7              ◆
4 REM  ◆      ｳｴﾃﾞｨﾝｸﾞ ﾚｰｽ          ◆
5 REM  ◆ Copyright by Y.Shiraki     ◆
6 REM  ◆     83/02/21   V.2         ◆
7 REM  ◆     CHAMPION SOFT          ◆
8 REM  ◆                            ◆
9 REM  ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
10 REM
20 SCREEN7,7:CLS:WIDTH40,25:COLOR 7,1
90 WIDTH 40,25
100 PLAY"T25504A+805D+2.D+8D4D804A+4A+805C4C804G4G8G4.G8F4D+8D+2.G8F4D+8D+2.R8R8
R4"
105 PLAY"T25504A+805D+2.D+8D4D804A+4A+805C4C804G4G8G4.G8F4D+8D+2.G8F4D+8D+2.R8R8
R4"
107 COLOR7
110 SYMBOL(40,24),"ｺﾝﾋﾟｭｰﾀ ﾀﾞｲｽﾌﾟﾚｲ ｹﾞｰﾑ",2,2,7
120 SYMBOL(32,64),"ｳｴﾃﾞｨﾝｸﾞ ﾚｰｽ",6,6,2
160 SYMBOL(350,160),"CHAMPION SOFT",2,2,7
169 COLOR 4,0
170 FORI=0TO6000:NEXTI:GOSUB11000:CLS
190 WIDTH40,25:COLOR5,0
200 LOCATE6,7:PRINT"1.ｹﾞｰﾑ ﾉ ﾙｰﾙ ｦ ﾐﾙ":LOCATE6,10:PRINT"2.ｹﾞｰﾑ ｦ ｽﾙ":LOCATE7,15:
PRINT"1.,2. ｲｽﾞﾚｶﾉ ｷｰｦ ｵｼﾃ ｸﾀﾞｻｲ"
210 P$=INKEY$
220 IFP$="1"THENCLS:GOTO300
230 IFP$="2"THENCLS:GOTO1000
240 GOTO210
300 LOCATE1,4:PRINT"ｺﾉ ｹﾞｰﾑ ﾊ":LOCATE1,6:PRINT"ｺﾉﾖ ﾆ ｳﾏﾚﾃｶﾗ ｲｶﾆﾊﾔｸ ｹｯｺﾝ ﾆ ｺﾞｰﾙｲﾝ
":LOCATE1,8:PRINT"ｽﾙｶ ｦ ｷｿｳ ｹﾞｰﾑ ﾃﾞｽ"
310 LOCATE1,10:PRINT"ﾒﾃﾞﾀｸ ｹｯｺﾝ ﾆ ｺﾞｰﾙｲﾝ ｼﾃ":LOCATE1,12:PRINT"ｳｴﾃﾞｨﾝｸﾞﾏｰﾁ ﾄ ﾄﾓﾆ
ﾃﾝｺﾞｸ ﾍ ｲｸｺﾄ ﾆ ﾅﾙｶ?"
320 LOCATE1,14:PRINT"ｿﾚﾄﾓ ﾊｻﾝ ﾉ ｳｷﾒﾆｱｲ ｼﾞｺﾞｸﾕｷ ﾄ ﾅﾙｶ?":LOCATE1,16:PRINT"ｻｯｿｸ ｳｴﾃ
ﾞｨﾝｸﾞ ﾚｰｽ ﾆ ﾁｮｳｾﾝｼﾃ ｸﾀﾞｻｲ"
330 GOSUB800
350 COLOR6:LOCATE5,1:PRINT"◆◆◆ ｹﾞｰﾑ ﾉ ﾙｰﾙ ◆◆◆"
360 COLOR5:LOCATE3,4:PRINT"1.ﾌﾟﾚｲﾔｰ ﾉ ﾆﾝｽﾞｳ":LOCATE5,5:PRINT"1-4 ﾆﾝ"
370 LOCATE3,7:PRINT"2.ｹﾞｰﾑ ﾉ ｼﾞｭﾝﾊﾞﾝ":LOCATE5,8:PRINT"ｽﾀｰﾄｼﾞ ﾆ ｹﾞｰﾑ ｦ ｽﾙﾋﾄ ﾉ ｼﾞｭ
```

```
ﾝﾊﾞﾝ ｦ":LOCATE5,9:PRINT"ｺﾝﾋﾟｭｰﾀ ﾆ ﾄｳﾛｸｼﾏｽ"
380 LOCATE3,11:PRINT"3.ｽﾀｰﾄｼﾞ ﾉ ﾃﾓﾁｼｷﾝ ﾊ 100 ﾏﾝｴﾝ ﾃﾞｽ"
390 LOCATE3,13:PRINT"4.ｹﾞｰﾑ ﾊ DICE CARD ｦ ﾋｸｺﾄﾆﾖｯﾃ ｽｽﾒﾏｽ":LOCATE5,14:PRINT"PLAY
ｺｰｽNO.75 ﾃﾞ ｺﾞｰﾙｲﾝ"
400 LOCATE3,16:PRINT"5.ｵﾓｲｶﾞｹﾅｲ ｼｭｳﾆｭｳ ﾔ ｼｼｭﾂ ｶﾞ ｱﾘﾏｽｶﾞ":LOCATE5,17:PRINT"ﾃﾓﾁｼｷﾝ
 ｶﾞ ﾏｲﾅｽ ﾆ ﾅﾙﾄ ﾊｻﾝ ﾃﾞ ｼｯｶｸ ﾆ":LOCATE5,18:PRINT"ﾅﾘﾏｽ"
410 GOSUB800
450 LOCATE3,4:PRINT"6.ｺﾞｰﾙｲﾝ ｽﾙﾄ ﾂｷﾞ ﾉ ﾎﾞｰﾅｽ ｶﾞ ﾃﾞﾏｽ":LOCATE6,5:PRINT"1 ﾁｬｸ.....
50 ﾏﾝｴﾝ":LOCATE6,6:PRINT"2 ﾁｬｸ.....30 ﾏﾝｴﾝ"
460 LOCATE6,7:PRINT"3 ﾁｬｸ.....10 ﾏﾝｴﾝ":LOCATE6,8:PRINT"4 ﾁｬｸ.....0"
470 LOCATE3,10:PRINT"7.ｹﾞｰﾑ ﾉ ｼｮｳﾊｲ":LOCATE5,11:PRINT"ｺﾞｰﾙ ﾆ ﾄｳﾁｬｸｼﾀﾄｷ ﾉ ﾃﾓﾁｼｷﾝｶ
ﾞ":LOCATE5,12:PRINT"ｵｵｲｼﾞｭﾝﾊﾞﾝ ﾆ ｷﾒﾏｽ"
480 GOSUB800:GOTO1000
800 LOCATE10,21:PRINT"ｽﾍﾟｰｽ ｷｰｦ ｵｼﾃ ｸﾀﾞｻｲ"
810 P$=INKEY$:IFP$<>" "GOTO810
820 CLS:RETURN
1000 REM *** ｶﾞﾒﾝ ***
1010 DIM K1(20),XD(20,20),YD(20,20),MK$(4),M$(4),LC(4),MY(4),Y(4),GN(4),HS(4),MU
$(6),ML$(6),RZ(6),M1$(4),M2$(4),CR$(2),SR(4)
1030 LINE(112,12)-(320,32),PSET,7,BF
1040 SYMBOL(120,16),"ｳｴﾃﾞｨﾝｸﾞ ﾚｰｽ",1.5,1.5,2
1050 CIRCLE(56,68),40,4,.44965
1100 X1=0:Y1=0:X0=528:Y0=48:RESTORE1130:FORK=1TO31:READX,Y:X=X*1.2*2:Y=Y*1.2
1110 IFX1=0THENX1=X:Y1=Y
1120 LINE(X0+X1,Y0+Y1)-(X0+X,Y0+Y),PSET,2:X1=X:Y1=Y:NEXTK
1130 DATA 17,7,15,4,13,2,10,1,7,2,5,3,3,5,2,8,1,10,1,12,2,15,3,17,4,20,8,25,13,3
0,17,32
1135 DATA 21,30,26,25,30,20,31,17,32,15,33,12,33,10,32,8,31,5,29,3,27,2,24,1,21,
2,19,4,17,7
1140 FORK=0TO3:LINE(152+K*96,48)-(184+K*96,132),PSET,6,B:NEXTK:LINE(436,46)-(476
,134),PSET,6,B
1150 FORK=0TO1:LINE(40+K*512,92)-(72+K*512,132),PSET,6,B:NEXTK
1160 FORK=0TO8:LINE(96,64+K)-(152,64+K),PSET,6:LINE(184,64+K)-(248,64+K),PSET,6:
LINE(280,64+K)-(344,64+K),PSET,6:LINE(376,64+K)-(436,64+K),PSET,6:LINE(476,64+K)
-(528,64+K),PSET,6:NEXTK
1200 FORK=0TO3:LINE(72,100+K*8)-(152,100+K*8),PSET,6:LINE(184,100+K*8)-(248,100+
K*8),PSET,6:LINE(280,100+K*8)-(344,100+K*8),PSET,6:LINE(376,100+K*8)-(436,100+K*
8),PSET,6:NEXTK
1300 REM *** ﾓｼﾞ ***
1310 C=3:XH=2:YV=2:Y0=56:X0=24:RESTORE60140:GOSUB50000:X0=60:RESTORE60010:GOSUB5
0000:COLOR6:LOCATE1,9:PRINT"ｽﾀｰﾄ"
1320 X0=532:RESTORE60080:GOSUB50000:X0=564:RESTORE60090:GOSUB50000:COLOR6:LOCATE
34,9:PRINT"ｺﾞｰﾙ"
1400 C=5:XH=1.4:YV=1.4:X0=158:RESTORE60160:FORK=0TO4:Y0=48+K*10:GOSUB50000:NEXTK
:X0=254:Y0=48:RESTORE60200:GOSUB50000:RESTORE60170:FORK=0TO3:Y0=57+K*10:GOSUB500
00:NEXTK
1410 X0=350:Y0=48:RESTORE60210:GOSUB50000:RESTORE60180:FORK=0TO2:Y0=61+K*13:GOSU
B50000:NEXTK
1420 XH=2:YV=2:X0=440:RESTORE60220:FORK=0TO1:Y0=48+K*32:GOSUB50000:NEXTK
1430 LINE(496,152)-(512,152),PSET,6:Y0=144:RESTORE60260:FORK=0TO1:X0=512+K*96:GO
SUB50000:NEXTK
1440 LINE(496,168)-(512,168),PSET,6:LINE(496,170)-(512,170),PSET,6:Y0=160:X0=512
:RESTORE60100:GOSUB50000:X0=544:RESTORE60170:GOSUB50000
1445 X0=576:RESTORE60220:GOSUB50000:X0=608:RESTORE60170:GOSUB50000
1450 COLOR2:LOCATE3,5:PRINT"0       7     19     38     50      75"
2000 REM *** ｹﾞｰﾑ ﾉ ｼﾞｭﾝﾋﾞ ***
2005 FORK=1TO4:MY(K)=100:NEXTK
2010 COLOR5:LOCATE8,19:PRINT"*ｹﾞｰﾑｦｽﾙ ﾆﾝｽﾞｳ ｦ":LOCATE8,20:PRINT"ｲﾝﾌﾟｯﾄ ｼﾃｸﾀﾞｻｲ(1-
4)"
2011 LOCATE9,22:INPUTNN
2012 IFNN<1THENLOCATE9,22:PRINTSPACE$(5):GOTO2011
2013 IFNN>4THENLOCATE9,22:PRINTSPACE$(5):GOTO2011
2015 M1$(0)="♠":M1$(1)="♥":M1$(2)="♦":M1$(3)="♣"
2016 M2$(0)="I":M2$(1)="O":M2$(2)="J":M2$(3)="K"
2018 COLOR3:LOCATE28,0:PRINT"ｺｰｽNO  MONEY"
2020 FORN=1TONN
2030 GOSUB10000:COLOR7:LOCATE1,18:PRINT"*";N;"ﾊﾞﾝﾒﾉﾋﾄ ﾉ ﾏｰｸ ｦ ｷﾒﾃ ｸﾀﾞｻｲ":GOSUB11
```

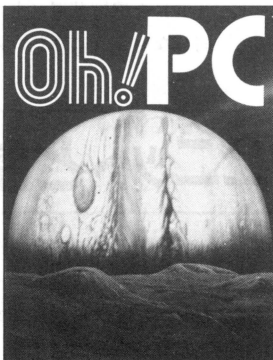
```
000
2040 COLOR7:FORK=0TO3:LOCATE6,20+K:PRINTM1$(K);" ﾅﾗ.....";M2$(K);" ﾉ ｷｰｦ ｵｽ":NEX
TK
2060 P$=INKEY$
2080 IFP$="I"THENM$="♠":GOTO2150
2085 IFP$="i"THENM$="♠":GOTO2150
2090 IFP$="O"THENM$="♥":GOTO2150
2095 IFP$="o"THENM$="♥":GOTO2150
2100 IFP$="J"THENM$="♦":GOTO2150
2105 IFP$="j"THENM$="♦":GOTO2150
2110 IFP$="K"THENM$="♣":GOTO2150
2115 IFP$="k"THENM$="♣":GOTO2150
2120 GOTO2060
2150 MK$(N)=M$:IFN=1GOTO2192
2170 FORJ=1TO N-1:IFMK$(J)=M$THENLOCATE3,18:PRINTSPACE$(30):LOCATE3,18:PRINT"***
 ﾏｰｸ ｶﾞ ﾁｮｳﾌｸ ｼﾃｲﾏｽ ***":GOTO2060
2190 NEXTJ
2192 COLOR2:LOCATE27,N:PRINTMK$(N):COLOR5:LOCATE28,N:PRINT"     0  100":C=6:XH=1:
YV=1:R=0:Y0=N*8:X0=606:RESTORE60280:GOSUB50000:X0=622:GOSUB50000:NEXTN
2300 FORK=1TONN:COLOR2:LOCATE3,11+K:PRINTMK$(K):NEXTK
2320 GOSUB10000
3000 REM *** ｹﾞｰﾑ ｶｲｼ ***
3010 LOCATE1,18:PRINT"* ｿﾚﾃﾞﾊ ｺﾚｶﾗ ｹﾞｰﾑ ｦ ﾊｼﾞﾒﾏｽ *":LOCATE3,20:PRINT"ｼﾞｭﾝﾊﾞﾝ ﾄ ﾏ
ｰｸ ｦ ﾏﾁｶﾞｴﾅｲﾖｳ":LOCATE3,21:PRINT"ｷｵﾂｹﾃ ｸﾀﾞｻｲ":FORJ=0TO4000:NEXTJ
3020 GOSUB10000
3100 FORN=1TONN
3110 IFMK$(N)="♠"THENM$(N)="I":GOTO3200
3120 IFMK$(N)≐"♥"THENM$(N)="O":GOTO3200
3130 IFMK$(N)="♦"THENM$(N)="J":GOTO3200
3140 IFMK$(N)="♣"THENM$(N)="K":GOTO3200
3200 NEXTN
3400 N=0
3500 N=N+1
3502 IF ((HS(N)<>1)*(GN(N)<>1)) THENGOTO3510
3504 IF N=NN GOTO3400
3506 GOTO3500
3510 COLOR7:LOCATE7,17:PRINT"*";N;"ﾊﾞﾝﾒﾉﾋﾄ DICE CARD":LOCATE9,18:PRINT"ｦ ﾋｲﾃ ｸﾀﾞ
ｻｲ"
3512 IFGN(N)=1GOTO6020
3514 IF Y(N)=1 THENY(N)=0:LOCATE7,17:PRINT"*";N;" ﾊﾞﾝﾒﾉﾋﾄ":LOCATE9,18:PRINT"ｺﾝｶｲ
ﾊ ｵﾔｽﾐﾃﾞｽ":GOTO6000
3520 FORK=1TO6
3522 R=RND(82)*100:R=INT(R)
3524 IFR<16THENRZ(K)=1:GOTO3550
3526 IFR<34THENRZ(K)=2:GOTO3550
3528 IFR<50THENRZ(K)=3:GOTO3550
3530 IFR<66THENRZ(K)=4:GOTO3550
3532 IFR<83THENRZ(K)=5:GOTO3550
3534 RZ(K)=6
3550 NEXTK
3552 LOCATE8,21:PRINT"(1-6)ﾉｷｰ ｦ ｵｼﾃｸﾀﾞｻｲ"
3555 P$=INKEY$
3556 IFP$="1"THENV=1:GOTO3570
3558 IFP$="2"THENV=2:GOTO3570
3560 IFP$="3"THENV=3:GOTO3570
3562 IFP$="4"THENV=4:GOTO3570
3564 IFP$="5"THENV=5:GOTO3570
3566 IFP$="6"THENV=6:GOTO3570
3568 GOTO3555
3570 LC=RZ(V):PX=V:PY=0
3701 IF LC(N)=60THENLC=1:GOTO3710
3702 IF ((LC(N)<60)*((LC(N)+LC)>60)) THENLC=60-LC(N):GOTO3710
3703 IF (LC(N)=50)THENLC=51-LC(N):GOTO3710
3704 IF ((LC(N)<38)*((LC(N)+LC)>37)) THENLC=38-LC(N):WW=3:GOTO3710
3705 IF ((LC(N)<19)*((LC(N)+LC)>18)) THENLC=19-LC(N):WW=2:GOTO3710
3706 IF ((LC(N)<7)*((LC(N)+LC)>6)) THENLC=7-LC(N):WW=1:GOTO3710
3710 LZ=RZ(V):IFLC(N)=50THENLZ=PX
```

```
3720 GOSUB16000:COLOR5:LOCATE10,20:PRINT"DICE CARD=":COLOR2:LOCATE22,20:PRINTLZ
3740 IFVZ=0GOTO3800
3750 ON VZ GOTO3755,3760,3765
3755 VZ=0:IF ((LC=1)+(LC=6)) THENGOTO3910
3757 GOSUB12500:GOTO6000
3760 VZ=0:IF ((LC<>1)*(LC<>4)) THENGOSUB12500:GOTO6000
3762 GOSUB12000:MY(N)=MY(N)+2000:COLOR5:LOCATE34,N:PRINT USING"####";MY(N):GOTO6
000
3765 VZ=0:IF ((LZ=1)+(LZ=5)+(LZ=6)) THENSR=1:SR(N)=1:FORH=0TO1:LINE(476,92+N*8+H
*2)-(552,92+N*8+H*2),PSET,6:NEXTH:GOTO3800
3766 SR=2:SR(N)=2:LINE(476,92+N*8)-(552,92+N*8),PSET,6
3800 IFLC(N)+LC<74GOTO4000
3810 LC=74-LC(N):IFLC=0GOTO3825
3820 GOSUB13000
3825 GOSUB10000
3830 RESTORE61840:READD1$,D2,D3$,D4:LOCATE8,17:PRINT"*":FORH=1TOD4:LOCATE9,16+H:
READD$:PRINTD$:NEXTH:GOSUB16000
3840 COLOR7:LOCATE7,22:PRINT"ｽﾍﾟｰｽ ｷｰｦ ｵｼﾃ ｸﾀﾞｻｲ"
3850 P$=INKEY$:IFP$<>" "GOTO3850
3860 GOSUB10000:VZ=1:GOTO3510
3910 GOSUB11000:LC(N)=LC(N)+1:COLOR5:LOCATE29,N:PRINTSPACE$(4):LOCATE29,N:PRINT
USING"####";LC(N):FORJ=0TO50:NEXTJ:LOCATE0,22:PRINT" "
3922 GOSUB10000
3924 RESTORE61850:READD1$,D2,D3$,D4:COLOR2:LOCATE8,17:PRINT"*":FORH=1TO D4:LOCAT
E9,16+H:READD$:PRINTD$:NEXTH:FORHH=0TO1000:NEXTHH:GOSUB16000
3930 NE=NE+1:GN(N)=1:AA=NE+HS:MX=(7-2*NE)*10:IFNE=4THEN MX=0
3940 MY(N)=MY(N)+MX:COLOR5:LOCATE34,N:PRINT USING"####";MY(N)
3950 GOSUB7000:COLOR2:LOCATE34,11+N:PRINTMK$(N):LOCATE35,11+N:PRINT USING"#";NE:
LOCATE36,11+N:PRINT"ﾁｬｸ"
3955 GOSUB10000
3960 GOSUB12700
3962 IF NN=AA THENGOTO3970
3966 GOTO6030
3970 FORH=0TO6:LOCATE10,17+H:PRINTSPACE$(22):NEXTH:LOCATE12,20:PRINT"*** GAME OV
ER ***"
3972 COLOR5:LOCATE9,22:PRINT"ﾓｳｲﾁﾄﾞ ﾔﾘﾏｽｶ ?(YES=Y,NO=N)"
3975 P$=INKEY$
3976 IFP$="Y"THENCLEAR:CLS:GOTO1000
3977 IFP$="y"THENCLEAR:CLS:GOTO1000
3978 IFP$="N"THENLOCATE1,1:END
3979 IFP$="n"THENLOCATE1,1:END
3980 GOTO3975
4000 IFLC(N)+LC<>67GOTO4100
4010 GOSUB13000
4020 GOSUB10000
4030 RESTORE61770:READD1$,D2,D3$,D4:COLOR2:LOCATE7,17:PRINT"*":FORH=1TO D4:LOCAT
E8,16+H:READD$:PRINTD$:NEXTH:GOSUB16000
4040 LOCATE7,22:PRINT"ｽﾍﾟｰｽ ｷｰｦ ｵｼﾃ ｸﾀﾞｻｲ"
4050 P$=INKEY$:IFP$<>" "GOTO4050
4060 GOSUB10000
4070 VZ=2:GOTO3510
4100 IF ((LC(N)<50)*(LC(N)+LC>49)) THENGOTO4120
4110 GOTO4300
4120 LC=50-LC(N):LOCATE0,22:PRINT" "
4130 GOSUB13000
4135 LOCATE0,22:PRINT" "
4140 LOCATE25,11+N:PRINT"  ":LINE(376,100+(N-1)*8)-(436,100+(N-1)*8),PSET,6:GOSU
B13500:COLOR2:LOCATE28,11+N:PRINTMK$(N):GOSUB13500
4150 GOSUB10000
4160 COLOR3:LOCATE8,17:PRINT"*ｼﾝﾛ ｾﾝﾀｸ":COLOR7:LOCATE9,19:PRINT"PLAY CARD ｦ ﾋｸ"
4170 LOCATE7,23:PRINT"ｽﾍﾟｰｽ ｷｰｦ ｵｼﾃ ｸﾀﾞｻｲ"
4172 COLOR6:LOCATE9,20:PRINT"1.5.6...ﾀﾞｲｶﾞｸｼﾝｶﾞｸ":COLOR5:LOCATE9,21:PRINT"2.3.4.
..ｼｭｳｼｮｸ"
4175 P$=INKEY$:IFP$<>" "GOTO4175
4180 GOSUB10000
4190 VZ=3:GOTO3510
4300 IF ((WW=1)+(WW=2)+(WW=3)) THENGOTO4320
```

```
4310 GOTO4500
4320 GOSUB13000
4330 GOSUB10000
4340 GOSUB11000:COLOR6:LOCATEWW*6,11+N:PRINT"    ":LINE(88+96*(WW-1),100+(N-1)*8)-
(152+96*(WW-1),100+(N-1)*8),PSET,6:COLOR2:LOCATE4+WW*6,11+N:PRINTMK$(N):GOSUB150
00:GOSUB13500
4350 IFWW=1THENRESTORE61070:GOTO4400
4360 IFWW=2THENRESTORE61190:GOTO4400
4370 IFWW=3THENRESTORE61380:GOTO4400
4400 WW=0:READD1$,D2,D3$,D4:COLOR7:LOCATE7,17:PRINT"*":FORH=1TO D4:LOCATE8,16+H:
READD$:PRINTD$:NEXTH:LOCATE8,19:PRINTD3$;" ﾏﾝｴﾝ"
4410 MY(N)=MY(N)+VAL(D3$):COLOR5:LOCATE34,N:PRINT USING"####";MY(N):GOSUB13500:G
OTO6000
4500 KD=1:KS=1:IFLC(N)=46THENY(N)=0
4501 IFCG=99THENCG=0:GOTO4503
4502 CG=0
4503 K=0:COLOR5
4504 LC(N)=LC(N)+KD:GOSUB11000
4505 COLOR5:LOCATE29,N:PRINTSPACE$(4):LOCATE29,N:PRINT USING"####";LC(N)
4506 FORJ=0TO50:NEXTJ:K=K+KD
4508 IFK<>LC THENGOSUB13500:GOTO4504
4509 GOSUB13500
4511 CX=LC(N)
4512 IF((CX>0)*(CX<7))THENPP=6:GOTO4517
4513 IF((CX>7)*(CX<19))THENPP=13:GOTO4517
4514 IF((CX>19)*(CX<38))THENPP=19:GOTO4517
4515 IF((CX>38)*(CX<50))THENPP=25:GOTO4517
4516 IF((CX>50)*(CX<75))THENPP=31:GOTO4517
4517 COLOR6:LOCATEPP-3,11+N:PRINT" ":LOCATEPP,11+N:PRINTMK$(N):GOSUB15000
4518 IFCX=19THENGOSUB11000:LOCATE13,11+N:PRINT"    ":LINE(184,100+(N-1)*8)-(248,1
00+(N-1)*8),PSET,6:COLOR2:LOCATE16,11+N:PRINTMK$(N):GOSUB13500
4519 IFCX=50THENGOSUB11000:LOCATE25,11+N:PRINT" ":LINE(376,100+(N-1)*8)-(436,100
+(N-1)*8),PSET,6:COLOR2:LOCATE28,11+N:PRINTMK$(N):GOSUB13500:GOTO4150
4528 GOSUB10000
4530 DX=LC(N)
4532 IFLC(N)>0THENLOCATE3,11+N:PRINT" "
4540 IFLC(N)<8THENRESTORE61010:GOTO4700
4550 IFLC(N)<20THENRESTORE61080:GOTO4700
4560 IFLC(N)<39THENRESTORE61200:GOTO4700
4570 IFLC(N)<51THENRESTORE61390:GOTO4700
4580 IFLC(N)>60THENRESTORE61710:GOTO4700
4590 IFSR=1THENRESTORE61510:GOTO4700
4600 RESTORE61610
4700 IFDX=67THENCOLOR6:GOTO4704
4701 IFDX=75THENCOLOR6:GOTO4704
4703 COLOR7
4704 READD1,D2,D3$,D4
4705 IFDX<>D1 THENFORH=1TO D4:READD$:NEXTH:GOTO4700
4710 COLOR7:LOCATE7,17:PRINT"*":FORH=1TOD4:LOCATE8,16+H:READD$:PRINTD$:NEXTH
4711 IFD3$="0"GOTO4720
4712 LOCATE15,21:PRINTD3$;" ﾏﾝｴﾝ"
4720 IFD3$="0"GOTO4740
4730 MY(N)=MY(N)+VAL(D3$)
4731 IFMY(N)>=0THENGOTO4737
4732 GOSUB12600:GOSUB7000:COLOR2:LOCATE34,11+N:PRINT"* ﾊｻﾝ":HS=HS+1:HS(N)=1
4733 GOSUB14000:FORT=0TO1000:NEXTT
4734 GOSUB10000
4735 IFNN=1THENNE=0:HS=1
4736 AA=NE+HS:GOTO3962
4737 GOSUB14000
4740 CG=0:LC=D2:IFD2=0GOTO6000
4750 IFD2<0THENCG=99:GOTO6000
4760 IFD2>0THENCG=999
6000 COLOR7:LOCATE8,22:PRINT"ｽﾍﾟｰｽ ｷｰｦ ｵｼﾃ ｸﾀﾞｻｲ"
6010 P$=INKEY$:IFP$<>" "GOTO6010
6012 IFCG=99THENKD=-1:KS=-1:GOTO4501
6013 IFCG=999THENKD=1:KS=1:GOTO4501
```

```
6030 GOSUB10000
6040 IF N<NN GOTO3500
6050 IF N=NN GOTO3400
7000 LOCATE32,11+N:PRINT" "
7010 ON SR GOTO 7100,7200
7100 FORH=0TO1:LINE(476,92+N*8+H*2)-(552,92+N*8+H*2),PSET,6:NEXTH:RETURN
7200 LINE(476,92+N*8)-(552,92+N*8),PSET,6:RETURN
10000 FORH=17TO23:LOCATE0,H:PRINTSPACE$(30):NEXTH:RETURN
11000 PLAY"T230O5L8C":RETURN
12000 PLAY"O4L48CL48DL48EL48FL48GL48AL48BO5L48C":RETURN
12500 PLAY"O4L48CO3L48BL48AL48GL48FL48EL48DL48C":RETURN
12600 PLAY"T160O2A8R16A8R16A8R16F2.G8R16G8R16G8R16E4R2":RETURN
12700 PLAY"T200O4D4.G8.G8G2D4A8.F+8G2D4G8.B8O5D4O4B8.G8G4F+8.G8A2":RETURN
13000 K=0
13010 GOSUB11000:LC(N)=LC(N)+1
13012 COLOR5
13013 LOCATE29,N:PRINTSPACE$(4)
13014 LOCATE29,N:PRINT USING"####";LC(N)
13016 FORJJ=0TO50:NEXTJJ
13020 K=K+1
13030 IF K<>LC THENGOSUB13500:GOTO13010
13035 GOSUB13500:RETURN
13500 LOCATE38,22:PRINT" ":RETURN
14000 LOCATE34,N:PRINTSPACE$(4):COLOR5:LOCATE34,N:PRINT USING"####";MY(N):GOSUB1
3500:RETURN
15000 LINE(248,80)-(248,128),PSET,6:RETURN
16000 IFD1$="67"THENLOCATE0,21:PRINTSPACE$(31):RETURN
16010 IFD1$="74"THENLOCATE0,21:PRINTSPACE$(31):RETURN
16020 RETURN
50000 REM *** ｶﾝｼﾞ ***
50010 READKK
50020 FORJ=1TOKK:READK1(J):FORL=1TOK1(J)+1:READXD(J,L),YD(J,L):NEXTL:NEXTJ
50030 FORJ=1TOKK:FORH=1TOK1(J)
50040 X1=X0+XD(J,H)*XH:Y1=Y0+YD(J,H)*YV*.5
50050 X2=X0+XD(J,H+1)*XH:Y2=Y0+YD(J,H+1)*YV*.5
50060 LINE(X1,Y1)-(X2,Y2),PSET,C:NEXTH:NEXTJ:RETURN
60010 DATA 5,2,5,4,4,6,2,8,1,4,6,12,6,1,8,3,8,14,1,4,10,12,10,1,2,14,14,14
60080 DATA 12,2,4.5,2,2.5,5,4,6,2,2.5,4,2.5,8,6,7.5,1,5.5,6.5,6.5,8,1,4.5,7.5,4.
5,14,1,4,9,2,13,1,5,9,6,12,1,8,4,14,4
60085 DATA 1,11,2,11,7,1,9,7,13,7,1,9,10,9,14,2,9,10,13,10,13,14,1,9,14,13,14
60090 DATA 11,2,4,2,2,10,6,13,2,6,5,4,11,2,14,1,2,5,7,5,2,13,2,8,3,8,8,1,7,9,12,
6,1,8,5,14,5,1,9,3,14,8,1,8,10,8,14
60095 DATA 2,8,10,13,10,13,14,1,8,12,13,12,1,8,14,13,14
60100 DATA 3,1,3,6,13,6,4,9,2,8,6,7,8,5,11,2,14,2,8,6,12,12,14,14
60125 DATA 3,10,2,11,7,13,13,14,14,2,14,6,12,10,8,14,1,11,2,13,4
60140 DATA 14,1,4,2,4,4,1,2,4,6,4,1,2,6,6,6,1,2,8,6,8,1,2,10,2,14,2,2,10,6,10,6,
14,1,2,14,6,14,1,14,2,10,3,1,12,3,12,11,1,12,7,14,7
60145 DATA 1,10,5,10,11,1,9,11,14,11,4,7,2,9,2,7,8,9,8,7,14,2,7,11,10,13,14,14
60160 DATA 3,2,8,2,8,14,6,12,1,5,6,2,10,1,11,6,14,10
60170 DATA 8,1,4.5,2,5,4,1,7,2,7.5,4,1,11,2,10,4,1,2,4,2,6,2,2,4,14,4,13,6,2,4,6
,11,6,8,8,4,8,8,8.5,9,8.5,13,8,14,7,13,1,2,9,14,9
60180 DATA 10,1,2,4,7,4,1,5,2,5,14,1,5,4,2,10,1,5,5,7,7,1,10.5,2,10.5,3.7,1,8,3.
7,13,3.7
60185 DATA 1,9.5,5,7.5,7,1,11,5,13,7,2,11.5,6.5,10.5,9,6,14,2,9,7,10.5,9,14,14
60190 DATA 2,2,8,6,7,8,2,13,2,6,3,8,6,14,13
60195 DATA 8,1,4.5,2,5,4,1,7,2,7.5,4,1,11,2,10,4,1,2,4,2,6,2,2,4,14,4,13,6,2,4,6
,11,6,8,8,4,8,8,8.5,9,8.5,13,8,14,7,13,1,2,9,14,9
60200 DATA 4,1,3,4,3,8,2,3,4,13,4,13,8,1,3,8,13,8,1,8,2,8,14
60210 DATA 10,1,8,2,8,3.5,1,3,3.5,13,3.5,1,6,5,6,7,2,6,5,10,5,10,7,1,6,7,10,7,1,
3,8.5,3,14,3,3,8.5,13,8.5,13,14,12,13
60215 DATA 1,6,10,6,12,2,6,10,10,10,10,12,1,6,12,10,12
60220 DATA 11,1,10,2,8,5,1,8,5,8,11,1,13,2,11,5,1,8,5,14,5,1,11,5,11,11,1,8,7,13
,7,1,8,9,13,9,1,8,11,14,11
60225 DATA 1,3,2,5,4,3,2,6,5,6,5,9,2,11,2,2,11,11,14,14,14
60230 DATA 13,1,2,2,2,6,2,2,2,6,2,6,6,1,2,6,6,6,1,4,9,6,9,1,4,7,4,13,1,2,8,2,14,
1,1,15,6,12,1,10,2,7,6,2,9,3,13,3,7,10
60235 DATA 1,8,5,14,10,1,8,10,8,14,2,8,10,13,10,13,14,1,8,14,13,14
```

```
60245 DATA 1,12,7,12,10,1,7,10,14,10,1,9,10,8,12,1,12,10,13,12,1,4,2,6,4,5,2,6,6
,6,5,7,5,8,6,9,2,11,2,2,11,11,14,14,14
60260 DATA 12,1,5,2,5,4,1,2,4,8,4,1,3,6,3,9,2,3,6,7,6,7,9,1,3,9,7,9,1,5,9,5,14,1
,4,10,2,14,1,6,10,8,13,1,8,6,14,6
60265 DATA 2,11,2,11,7,9,14,3,11,7,11,14,14,14,14,13,1,12,2,13,5
60270 DATA 18,1,1,3,6,3,1,5,3,5,14,1,2,5,5,5,1,2,7,5,7,1,2,3,2,11,1,1,12,5,9,1,8
.5,2,8.5,4,1,6,4,11,4,1,7,4,7,7,1,10,4,10,7,1,6,7,14,7
60275 DATA 1,7,9,7,13,2,7,9,10,9,10,13,1,7,11,10,11,1,7,13,10,13,2,12,2,13,14,14
,14,1,14,8,10,14,1,13,3,14,5
60280 DATA 3,1,2,2,14,2,2,8,2,8,6,4,14,4,8,6,13,6,13,10,12,14,11,13
60290 DATA 4,1,2,2,2,14,2,2,2,14,2,14,14,1,8,2,8,8,1,2,8,14,8
60345 DATA 1,7,14,9,14,1,10,6,14,6,2,12,2,12,8,10,14,1,12,8,14,14,1,13,2,14,5
60385 DATA 1,5,11,5,14,2,5,11,12,11,12,14,1,5,14,12,14
60395 DATA 1,5,11,5,14,2,5,11,11,11,11,14,1,5,14,11,14
61010 DATA 1,0,-2,1,オミヤマイリ ニ イク.
61020 DATA 2,0,+3,2,ヨウチエン ニ ニュウエン,イワイキン ヲ モラウ.
61030 DATA 3,+3,0,2,シチコ゛サン ノ オイワイ.,3ツススム.
61040 DATA 4,-3,0,2,マイニチ オネショオシテ シカラレル.,3ツ モト゛ル.
61050 DATA 5,0,+3,2,ヨウチエン ヲ ソツエン.,オイワイ ヲ モラウ.
61060 DATA 6,0,-5,1,ヒ゜アノ ノ オケイコ ニ イク.
61070 DATA 7,0,+3,2,ショウカ゛ッコウ ヘ ニュウカ゛ク.,オイワイ ヲ モラウ.
61080 DATA 8,+1,0,4,2ネン ニ ナッテ,ソロハ゛ンシ゛ュク ヘ カヨウ.,シ゛ヨウタツスル.,1ツ ススム.
61090 DATA 9,0,-3,1,3ネン ニ ナッテ ショト゛ウ ニ カヨウ.
61100 DATA 10,0,0,1,シ゛テンシャ ニ ノレルヨウ ニ ナル.
61110 DATA 11,+3,0,2,ソロハ゛ン 3キュウ コ゛ウカク,3ツ ススム.
61120 DATA 12,0,+3,2,ヒ゜アノ ノ ハッヒ゜ヨウカイ,オイワイ ヲ モラウ.
61130 DATA 13,0;-1,1,ダ゛イスキ ナ オモチャ ヲ カウ.
61140 DATA 14,0,-5,1,カ゛クシュウシ゛ュク ヘ カヨウ.
61150 DATA 15,-5,0,2,ハツコイ テ゛ セイセキカ゛ オチル.,5ツ モト゛ル.
61160 DATA 16,+3,0,3,カ゛クシュウシ゛ュク ノ ,シ゛ツリョクテスト テ゛ 3ハ゛ン,3ツ ススム.
61170 DATA 17,0,-3,2,ショウカ゛ッコウ 6ネン,シュウカ゛クリョコウ ニ イク.
61180 DATA 18,0,+3,2,ショウカ゛ッコウ ヲ ソツキ゛ョウ,オイワイヲ モラウ.
61190 DATA 19,0,+3,2,チュウカ゛ッコウ ヘ ニュウカ゛ク,オイワイ ヲ モラウ.
61200 DATA 20,0,-2,1,テニス ノ ラケット ヲ カウ.
61210 DATA 21,0,-3,1,エイカイワ ヲ ナラウ.
61220 DATA 22,+3,0,2,ショト゛ウテン ニ ニュウセン.,3ツ ススム.
61230 DATA 23,0,-8,1,ラシ゛カセ ヲ カウ.
61240 DATA 24,0,+30,1,クイス゛ ニ アタル.
61250 DATA 25,0,-10,1,シンユウ ト リョコウ ニ イク.
61260 DATA 26,0,0,1,ニキビ゛ テ゛ ナヤム.
61270 DATA 27,0,0,2,アメリカシ゛ン ノ,ヘ゜ンフレント゛ カ゛ テ゛キル.
61280 DATA 28,-5,0,3,キマツシケンテ゛,カンニング゛ カ゛ ミツカル.,5ツ モト゛ル.
61290 DATA 29,-3,0,2,シンユウ ト オオケ゛ンカ.,3ツ モト゛ル.
61300 DATA 30,0,-5,2,チュウカ゛ッコウ 3ネン,シュウカ゛クリョコウ ヘ イク.
61310 DATA 31,0,-2,1,カ゛ッコウ ノ カ゛ラスマト゛ ヲ ワル.
61320 DATA 32,0,+15,2,マイコン ヲ カウタメ,オコス゛カイ ヲ モラウ.
61330 DATA 33,0,+5,2,クラブ゛ ノ カ゛ッシュク ノ タメ,コス゛カイ ヲ モラウ.
61340 DATA 34,+3,0,2,キマツシケン テ゛ 1ハ゛ン,3ツ ススム.
61350 DATA 35,0,-2,2,シンユウ カ゛ ヒ゛ヨウキ,ミマイニ イク.
61360 DATA 36,-1,0,3,ヘアスタイル ヲ オトナッホ゜ク カエテ,センセイ ニ シカラレル.,1ツ モト゛ル.
61370 DATA 37,0,+5,2,チュウカ゛ッコウ ヲ ソツキ゛ョウ,オイワイ ヲ モラウ.
61380 DATA 38,0,+5,2,コウコウ ニ ニュウカ゛ク.,オイワイ ヲ モラウ.
61390 DATA 39,0,0,1,コウコウ テ゛ テニスフ゛ ニ ハイル.
61400 DATA 40,0,0,2,カ゛ールフレント゛(ホ゛ーイフレント゛)ト,ハシ゛メテ テ゛ート.
61410 DATA 41,0,+20,3,マイコンフ゜ロク゛ラムコンテスト ニ,ニュウショウ.,ショウキン ヲ モラウ.
61420 DATA 42,0,-8,1,サイクリンク゛シャ ヲ カウ.
61430 DATA 43,0,0,1,タイイクサイ.
61440 DATA 44,0,0,1,カ゛クエンサイ.
61450 DATA 45,-3,0,3,シ゛ツリョクテスト テ゛,セイセキ カ゛ サカ゛ル.,3ツ モト゛ル.
61460 DATA 46,0,-10,2,スキーテ゛ ケカ゛ ヲ シテ,ニュウイン.
61470 DATA 47,0,0,2,カ゛ールフレント゛(ホ゛ーイフレント゛)カ゛,シ゛タク ニ アソヒ゛ニクル.
61480 DATA 48,-3,0,4,カ゛ールフレント゛(ホ゛ーイフレント゛),ノコトカ゛ キニナッテ,ヘ゛ンキヨウ カ゛ テニツカナイ.,3ツ モト゛ル.
61490 DATA 49,+1,0,3,シ゛ツリョクテスト ノ セイセキ カ゛,スコシアカ゛ル.,1ツ ススム.
61500 DATA 50,0,0,3,シンロ センタク,1.5.6...タ゛イカ゛クシンカ゛ク,2.3.4...シュウショク.
61510 DATA 51,0,0,1,タ゛イカ゛ク ヲ シ゛ュケン.
61520 DATA 52,0,+5,2,タ゛イカ゛クコ゛ウカク,オイワイ ヲ モラウ.
61530 DATA 53,0,-15,1,シ゛ト゛ウシャ ノ メンキョ ヲ トル.
```

```
61540 DATA 54,0,+10,2,ｶﾃｲｷｮｳｼ ﾉ,ｱﾙﾊﾞｲﾄﾘｮｳ ｦ ﾓﾗｳ.
61550 DATA 55,0,-50,1,ﾊﾜｲ ﾆ ｶｲｶﾞｲﾘｮｺｳ.
61560 DATA 56,0,+5,2,ｾｲｾｷﾕｳｼｭｳ,ｼﾖｳｶﾞｸｷﾝ ｦ ﾓﾗｳ.
61570 DATA 57,-2,0,3,ｺｲﾋﾞﾄｶﾞ ﾃﾞｷ,ﾍﾞﾝｷｮｳ ｶﾞ ｵﾛｿｶﾆﾅﾙ.,2ﾂ ﾓﾄﾞﾙ.
61580 DATA 58,0,+50,2,ﾏｲｺﾝﾌﾟﾛｸﾞﾗﾑｺﾝﾃｽﾄ,ｻｲﾕｳｼｭｳｼﾖｳ ﾆｭｳｾﾝ.
61590 DATA 59,0,0,1,ﾀﾞｲｶﾞｸ ｿﾂｷﾞｮｳ.
61600 DATA 60,0,0,2,ﾄﾞｺﾉ ｶｲｼｬﾆ ﾆｭｳｼｬｽﾙｶ,ﾏﾖｳ.
61610 DATA 51,0,0,2,ｼｭｳｼｮｸ ﾆ ﾕｳﾘ ﾅ,ｺﾝﾋﾟｭｰﾀｶﾞｸｲﾝ ﾆ ﾆｭｳｶﾞｸ.
61620 DATA 52,0,0,2,ｺﾝﾋﾟｭｰﾀｶﾞｸｲﾝ ｦ ｿﾂｷﾞｮｳ.,ｼｭｳｼｮｸｻｷｶﾞ ｷﾏﾙ.
61630 DATA 53,0,-15,1,ｼﾞﾄﾞｳｼｬﾉ ﾒﾝｷｮ ｦ ﾄﾙ.
61640 DATA 54,0,-20,1,ｽﾃﾚｵ ｦ ｶｳ.
61650 DATA 55,0,+20,2,ﾊｼﾞﾒﾃ ﾉ ﾎﾞｰﾅｽ ｦ,ﾓﾗｳ.
61660 DATA 56,0,-5,1,ｱﾀﾗｼｲ ｽｰﾂ ｦ ｶｳ.
61670 DATA 57,0,-30,2,ｸﾙﾏｦ ｶｲ,ｱﾀﾏｷﾝ ｦ ﾊﾗｳ.
61680 DATA 58,0,+10,1,ｸｲｽﾞ ﾆ ｱﾀﾙ.
61690 DATA 59,-5,0,3,ﾋﾞｮｳｷ ﾃﾞ ﾆｭｳｲﾝ,ｶｲｼｬ ｦ 1ｶｹﾞﾂ ﾔｽﾑ.,5ﾂ ﾓﾄﾞﾙ
61700 DATA 60,0,0,2,ﾎｶﾉｶｲｼｬ ｶﾗ ｽｶｳﾄｻﾚ,ｼﾝﾛ ﾆ ﾏﾖｳ.
61710 DATA 61,0,0,1,ｱﾀﾗｼｸ ｼｭｳｼｮｸ ｽﾙ.
61720 DATA 62,0,+13,2,ﾊｼﾞﾒﾃ ﾉ ｷｭｳﾘｮｳ ｦ,ﾓﾗｳ.
61730 DATA 63,0,0,1,ｺｲﾋﾞﾄ ｶﾞ ﾃﾞｷﾙ.
61740 DATA 64,0,0,1,ｺｲﾋﾞﾄ ﾄ ﾌｧｰｽﾄｷｽ ｦ ｽﾙ.
61750 DATA 65,0,-20,2,ｱﾀﾗｼｸ ｸﾙﾏｦｶｲ,ｱﾀﾏｷﾝ ｦ ﾊﾗｳ.
61760 DATA 66,0,+30,2,ｶｲｼｬ ﾉ ｷﾞｮｳｾｷｶﾞﾖｸ,ﾘﾝｼﾞﾎﾞｰﾅｽ ｦ ﾓﾗｳ.
61770 DATA 67,0,0,3,ﾀｶﾗｸｼﾞ ｦ ｶｳ.,PLAY CARD ｶﾞ,1ﾏﾀﾊ4...+2000 ﾏﾝｴﾝ.
61780 DATA 68,0,+2,2,ｷﾝﾑｾｲｾｷ ｶﾞ ﾖｸ,ﾄｸﾍﾞﾂ ｼｮｳｷｭｳ.
61790 DATA 69,-4,0,2,ﾁｺｸ ﾑﾀﾞﾝｹｯｷﾝ ｶﾞ ｵｵｲ.,4ﾂ ﾓﾄﾞﾙ.
61800 DATA 70,-4,0,2,ｴﾝﾀﾞﾝ ｶﾞ ﾊﾀﾞﾝ ﾆ ﾅﾙ.,4ﾂ ﾓﾄﾞﾙ.
61810 DATA 71,0,-1000,1,ｻﾗｷﾝ ﾉ ｼｬﾂｷﾝ ｶﾞ ｶｻﾑ.
61820 DATA 72,-4,0,2,ｺｲﾋﾞﾄ ﾄ ｹﾝｶｽﾙ.,4ﾂ ﾓﾄﾞﾙ.
61830 DATA 73,0,-100,1,ｹｯｺﾝ ﾉ ｼﾞｭﾝﾋﾞ ｦ ｽﾙ.
61840 DATA 74,0,0,3,ｹｯｺﾝ ﾉ ﾋﾄﾞﾘ ｷﾏﾙ.,PLAY CARD ｦ ﾋｸ,1.6...ｺﾞｰﾙｲﾝ.
61850 DATA 75,0,0,4,ｹｯｺﾝ ﾆ ｺﾞｰﾙｲﾝ.,1ﾁｬｸ....50 ﾏﾝｴﾝ,2ﾁｬｸ....30 ﾏﾝｴﾝ,3ﾁｬｸ....10 ﾏﾝ
ｴﾝ.
62010 PLAY"T25504A+805D+2.D+8D4D804A+4A+805C4C804G4G8G4.G8F4D+8D+2.G8F4D+8D+2.R8
R8R4"
```

フロッピーディスクユニット
―各社ドライブユニットの比較検討―

●山科　好

①はじめに

フロッピーディスク装置（注1）が初めて世に姿を現わしたのは，1960年，Laboratory for Electronics社による“BD-103”と言われているものだ。流動の激しいエレクトロニクスの世界では，ずいぶん昔のことである。データストレージ用のデバイスも，磁気デバイスを中心に，この時期から様々な物が発明・試作され始める。BD-103は現在知られているフロッピーディスク装置（以下ＦＤＤと略記する）とは，似ても似つかぬものであった。

1972年9月，突然IBM社はIBM-3740データEntry Systemを発表した。これに採用されたＦＤＤはＩＢＭ-33ＦＤと呼ばれ，従来のパンチカードによるデータ作成を革命的に変革するデバイスで，大反響を呼び起こした。

この33ＦＤが発表されるやいなや，各独立ペリフェラルメーカーは競ってこのタイプのＦＤＤの生産を開始し，この時期をもってフロッピーディスク時代の真の幕開けとなる。1975年項から普及しはじめた33ＦＤは，11年経過した現在でもすべての８インチスタンダードとして“3740フォーマット”の名称で変わらぬ地位を保っている。

さて33ＦＤは1976年に43ＦＤ，77年に53ＦＤと改良が加わり，メモリ容量も２倍，４倍と増大しているが，５インチの場合はどうであろうか。

BD-103の概略	
メモリ容量	0.12Mビット
トラック数	40 track
回転速度	1800～8000 rpm

ＩＢＭ-33ＦＤの概略	
メモリ容量	3.1Mビット
トラック数	76 track
セクタ数/トラック	26セクタ
回転速度	360 rpm
記録密度	3200ＢＰＩ
転送速度	250Ｋ bps

５インチ，正確に言うならばミニフロッピー（注２）はスタンダード版８インチＦＤＤに対して，米国シュガート・アソシエイツが開発したものである。『カセットデータレコーダ並みのコンパクトさと，ＦＤＤなみの性能』をスローガンに，1976年９月にSA-400の型番で発表された。スタンダードＦＤＤの信頼性と容量を落とし，低コスト化を実現させたものであるが，このミニ版ＦＤＤもスタンダード版のそれと同様，大容量高速化，高信頼性を追求されて現在に至っている。現在ではYEデータ社の約1Mバイト（フォーマット時），シーク速度3msといった最新の８インチＦＤＤと変わらぬ物まで出現している。

（注１）　ディスケットとはＩＢＭ社の呼称で，正確には“The IBM Diskette for Standard Data Interchange”であり，ＩＳＯ，ＪＩＳでは「磁気フレキシブルディスクカートリッジ」（Flexible Disk Cartridge）と名称を設定している。

（注２）　５インチのミニフロッピー（Mini Floppy），ミニディスケット（Mini Diskette）とはシュガート社の商標である。くれぐれも磁気メディア（ディスケット，ディスク，シート）を機械装置（ドライブユニット）と混同しないようにされたい。

②５インチドライブユニットと制御方法

さて注目の５インチＦＤＤだが，最近の販売価格の低廉化には目を見張るものがある。エレクトロニクス製品の世界では，この手の低価格化競争は日常茶飯事とはいえ，一昨年あたりからシステムハウスやガレージメーカーの新規参入が増えたため，低廉化競争の真っ盛りといった感すら持つ。もちろんこういった状況は，エンドユーザーにとって願ってもないことは言うまでもない。

ＦＤＤ装置はメインメモリの少ない８ビット機にとって，ランダムアクセス可能な高速大容量ストレージであり，マイクロコンピュ

ータを使って実務的な処理を行わせるには，必需品の一つである。ワードプロセッサやデータ集計，ソフトウェア開発などの仕事を行う上で，なくてはならないデバイスであると言えよう。

さて，FM-8，7のFDDとの接続方法は，NECのPCシリーズなどとは異なり，非インテリジェントタイプの通常の設計方法となっている。しかしFDC(Floppy Disk Controller)用のLSIを含めた制御ユニットはCPU本体部の中ではなく，FDDユニットケース内に設けられている。したがって，FDCユニットまで延々とCPUバスラインを引き延ばさなければならない。もちろんバスラインそのままを何メートルも延長したりすれば，ノイズなどの混入によるCPUの誤動作，動作停止は必至である。そのためバスドライバ回路を設けることで，この問題を何とか回避している。

FM-8，7もこの理由により，バスバッファの小さなボックスないしカードを必要とする。

表中でシーク速度は，リニアモータを使った特異なヘッド駆動メカニズムを持つエプソン社のSD-320が多少遅い。これは米国Per-Sci社の物と比べ，ヘッドメカニズム自体が重いためかもしれないが，ヘッド移動の際，ほとんど騒音を発生しないので快い。IBM-33FDからの従来の方式を採用したYEデータ社のＹＤ-274(ＭＢ27601に使用)のスクリュードライブ方式や，ＦＤ-55Ｂ，ＪＡ-551のスチールベルトドライブ方式の騒々しさに閉口している読者も，一度動作を目前で確認されたい。シークの遅速な点を指摘したが，従来の物，たとえば前述のＹＤ-274が20msecであった点に比べると，低騒音という利点で相殺されよう。551，55Ｂ両者間ではＴＥＡＣ製の55Ｂの方が飛び抜けて騒々しいが，551にはヘッドアクチュエータがないためである。メカニズムの信頼性やパーツの品位は55Ｂが上等である。

またモータのセットリングタイムなどは，FMシリーズの場合DOSのブートアップ後，モータは回転し始めたら電源を切るまで回り続けるので無関係なようである。(LFD-550/FMは内部タイマによりモータのON/OFF動作を行う)

さてもう一方の，ユニット全体の性能決定要因である電源部はどうであろうか。

定電圧電源方式には，スイッチング方式とシリーズレギュレータ方式の２つがある。前

③各社FDDユニットと接続方法

今回，調査の対象とした機種は表１のとおりであり，オプションの接続ケーブル，バスバッファボックス（名称はおのおの異なる）なども含めてある。ＦＤＤユニットとFM-8，７との接続方法は数通り考えられるが，現在市販されている機器を組み合せると図１のようになる。

ＦＭ-8の場合いずれの方法をとってもすべてのオペレーティングソフトウェアが動作可能である。またFM-7の場合も，クロック周波数は８MHz(CPUの動作周波数２MHz)であるが，ウエイトをかけて自動的にFM-8と同じクロックに切り換えてアクセスする設計になっている。

さてドライブ本体の方であるが，前述したようにＦＭ-8シリーズのＦＤＤユニットは，制御用のＦＤＣが載ったコントロール基板が組み込まれている。各社のＦＤＤにも図２のようなボードが組み込まれており，大方において回路上の差異は見られない。

まずＦＤＣであるが，このLSIはドライブユニットへのデータのリード／ライト，メカニカル部分の駆動や，各センサの検出などをいっさいこのチップのみで行う高機能な部品であり，制御全般を司る。各ドライブメーカーでは，ウエスタンデジタルがオリジナルの1791，もしくはコンパチブルな富士通の8876，8877を使用している。次にＲ／Ｗヘッドからの信号をクロックとデータに分離するＶＦＯ回路，そしてバスバッファ部から成り立っている。

ＦＤＤユニットはこのボードと電源，ドライブユニットの３点から構成されており，ボードは各社とも同一動作をするのでユニット自体の性能の差は，ドライブユニットと電源の能力の２点で決定する。とは言っても，ドライブユニット自体はエプソン自社製のSD-320を除き，他はすべて大手メーカーからの購入品である。また，そのユニットも松下通信工業製のＪＡ-551とTEAC社製のＦＤ-55Ｂの２種のいずれかしか使われていない。おのおのの性能比はドライブ自体に何を採用したかによる。表３に各ユニットの性能特徴を示す。

表１　調査機種内訳

型番／タイプ	製造会社名	バスドライバ	ケーブル	定　価
CF-80FM 8	㈱湘南電送機器（コマンドポスト）	〈オプション〉MFP-22603/ミニフロッピーアダプタ 15,000円	本体に付属	S 113,000 W 163,000 合計 178,000
LFD-550/FM	㈱東京電子科学機材	〈オプション〉FDインタフェースカード 33,000円	バスドライバに付属	W 148,000 合計 181,000
THIN LINE/FM	㈱亜土電子工業	FDDに内蔵	本体に付属	S 110,000 W 168,000 合計 168,000
TF-20 富士通FM-8専用機（FM-7も可）	㈱エプソン	〈オプション〉FDD I/Fセット ♯9010 15,000円	バスドライバに付属	W 163,000 合計 178,000
KD280/FM-8	㈱工人舎	〈オプション〉ミニFDアダプタ ADP-FM I 15,000円	〈オプション〉CBL-FM I 3,000円	S 125,000 W 188,000 合計 206,000
disk-80F（S）	㈱アイテム	〈オプション〉disk-I/O 17,000円	〈オプション〉CBL-3 7,000円	S 98,000 W 168,000 合計 192,000

*「S」はシングルドライブ，「W」はデュアルドライブを意味する。
*「合計」はオプション価格を加えた２ドライブFDDの定価である。
*なお，調査日付は２月１日である。

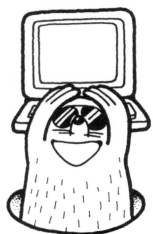
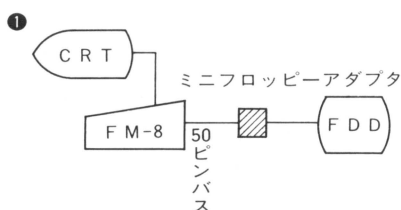
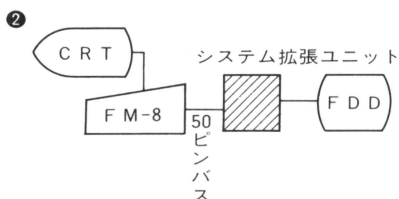
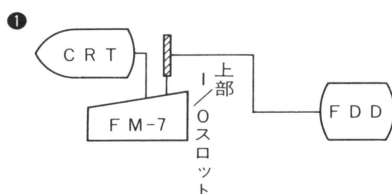
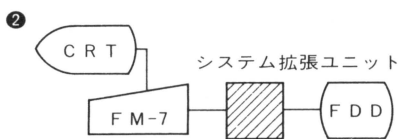
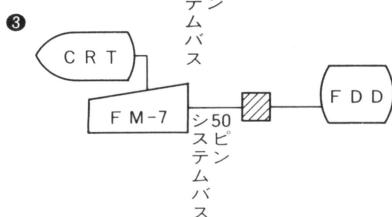



表2　ハードウェア明細

	速度（平均アクセス）	ドライブユニット	ヘッド移動メカニズム	電源ユニット	ＦＤＣチップ
CF-80FM	281ms	FD-55B（ＴＥＡＣ）	スチールベルト方式	スイッチングレギュレータ	8876（8866）
LFD-550/FM	281ms	JA-551（松下通工）	スチールベルト方式	スイッチングレギュレータ	
THIN LINE/FM	281ms	FD-55B（ＴＥＡＣ）	スチールベルト方式	スイッチングレギュレータ	8877
TF-20	281ms	SD-320（自社製）	リニアモータ方式	スイッチング／シリーズ併用	8876A
KD-280/FM-8	281ms	FD-55B（ＴＥＡＣ）	スチールベルト方式	シリーズレギュレータ	1791
disk-80F	281ms	JA-551（松下通工）	スチールベルト方式	シリーズレギュレータ	8877

＊カタログ等を見れば解る部分，また必要と思われない箇所はすべて省略した。

者は変圧トランスを含まない方式の場合きわめて小型軽量に仕上がる。また対入力変換効率も60％～80％程度で，効率は高いが，リップル特性が多少犠牲となる難点を持つ。リップル特性を高めようとすると高周波数帯でスイッチ動作を行うため，外部に多量の高周波ノイズを散乱させる結果となる。

逆に後者では変圧トランスにより，２次電圧を出力電圧近くまでドロップさせ安定化させる。そのためノイズの心配は不要であり，リップル特性も良好となり得る。が，反面，重量のあるトランスを使う，ユニット容積の割に出力がとれず効率が悪い，などの難点がある。特に心配なのは，ＦＤＤ装置に強力な磁力線を発生するパワートランスを組み込むことで，これらの磁力線が不完全な電磁シールドから漏れて，ＣＲＴやディスケットに悪影響をおよぼす可能性がある，ということである。なお，このタイプの電源がいちばん低コストで製作できる点を明記しておく。

データストレージとしてのマイクロFDDの現状

標準（８インチ）ＦＤＤを小型化することで生まれたミニ（５インチ）ＦＤＤであったが，さらに取り扱いやすくするために製造されたのがマイクロＦＤＤである。

このマイクロＦＤＤは，コンピュータの世界では珍しく国産メーカーの主導によって生み出されたが，残念なことに，二つの形式を生み出してしまった。日立・マクセル系グループによる３インチ方式と，ソニーによる3.5インチ方式で，両者間の互換性はない。

米国の独立ＦＤＤメーカーの一部は両者に協調したが，独自のフォーマットを打ち出した米国メーカーもある。

ＦＤＤ製造の大手である松下通信工業は，最近，日立方式を採用して本格生産を開始したが，いずれにせよ，３インチ/3.5インチのメディアはＶＴＲの規格と同様，二つの方式が並存してしまった。こういった状況で一番迷惑するのは，我々エンドユーザーであるといえよう。（Ｙ．Ｙ．）

エプソン

工人舎

ADO

コマンドポスト

東京電子

アイテム

④総合評価として

繰り返すようだが，どのメーカーのセットも，動作はすべて完全で，FLEX® もOS-9™ もCP/M® も満足に動作する。細かな点で比較をするならば，ＦＤＤユニットと電源の能力で多少，相違がみられる。それも最終的にはユーザーの好みの問題であり，不良動作が発生する，といったレベルの問題ではない。動作速度も非インテリジェントタイプである以上，オペレーティングソフトウェア，たとえばDisk BASIC 上のＦＤＣ制御ルーチンのタイマカウンタあたりの数値を入れ換えない限り，すべて，平均アクセス281msで一律に変化はない。いくらドライブユニット自体が79msまで上がろうとも,OSによって281msの動作を強いられる。

ケースの堅牢さについては，しっかりした物の方が良いに決っている。通信工業用や軍用のセットではないので，外観デザインと相反する点もあり，これは読者の判断におまかせしたい。ただ，ケース内部の処理についてであるが，塗装はおろかメッキ処理すらしていない物も数台あった。低コストを極限まで追求する最近の一般向けコンピュータ機器の中には，ケースをあけると驚くほどの造作となっているものがある。技術進歩の激しい世

表３　各ドライブユニット性能一覧

	EPSON：SD-320	松工：JA-551	TEAC：FD-55B
記憶容量 (1)アンフォーマット時(MFM) (2)フォーマット時(16セクタ/trk)(MFM)	500KBytes 328KBytes	500KBytes 328KBytes	500KBytes 328KBytes
記憶密度(side 1, trk 39)	5876 BPI	5876 BPI	5876 BPI
転送速度(MFM)	250K bits/sec	250K bits/sec	250K bits/sec
トラック内平均回転待ち時間	100msec	100msec	100msec
アクセス時間 (1)トラック間(シーク速度) (2)ランダムトラック間移動平均 (3)セトリング時間	15msec 220msec 15msec	6msec 79msec 15msec	6msec 93msec 15msec
ヘッドロード時間	35msec	(50msec)	50msec
モータ起動時間	0.5sec	1.0sec	0.4sec
モータ回転数	300rpm	300rpm	300rpm
トラック密度	48TPI	48TPI	48TPI
トラック総数	80	80	80
R/Wヘッド位置決め方式	ボイスコイルモータ(フローティング一体化)	スチールベルト/ステッピングモータ	スチールベルト/ステッピングモータ
主軸モータ/駆動方式	アウターロータ型ブラシレストランジスタサーボモータ/ベルトドライブ	FGサーボDCモータ/シャフト直結	FGサーボDCモータ/シャフト直結
動作時環境 (1)温度 (2)湿度 (3)最大湿球温度 (4)振動	4℃～43℃ 20%～80% RH 29℃/H 5～60Hz　0.6G以下	5℃～46℃ 20%～80% RH 15℃/H 2～50Hz　0.5G以下	4℃～46℃ 20%～80% RH 29℃/H 2～50Hz　0.25G以下
電源 (1)＋12V(±５%・リップル除く) (2)＋ 5V(±５%・リップル除く)	0.7A(typ)(リード/ライト時) 0.25A(typ)	0.6A(typ) 0.7A(typ)	0.5A(typ) 0.55A(typ)
外形寸法(パネルを除く) (1)幅 (2)奥行き (3)高さ	146mm 235.5mm 28.0mm	145mm 213mm 41.3mm	146mm 203mm 41.3mm
重量	1.35kg	1.5kg	1.5kg
信頼性 (1)MTBF (2)MTTR (3)設計寿命 (4)ソフトエラーレイト (5)ハードエラーレイト (6)シークエラーレイト	10000H(POH) 30m 5年 10^9 ビットに1回以下 10^{13}ビットに1回以下 10^6 シークに1回以下	10000H 30m 5年 10^9ビット(リトライ2回) 10^{12}ビット 10^6シーク回	10000H 30m 5年 10^9 ビット 10^{12}ビット 10^6シーク回

界であるから，こういう物を10年，20年も使用する人はまずいないと考えるが，内部をあけるとフレームや金具に錆がでて赤くなってしまうのは，あまり気分は良くないだろう。

対埃性についてであるが，これはセットの発熱の度合いと関連する。発熱が少なければケース上面や両サイドの通気孔を少なくできる。外部との開口面積が少なくなるわけである。当然，チリや埃の内部混中も減るであろう。

反対に発熱が多いと通気孔が多くなり，それだけ内部の磁気ヘッドやディスケットが汚れてしまう。

最後に，富士通はDisk BASIC F-BASIC Version 1.0，3.0を別売していない。このためコンパチディスクメーカーは，おのおの独自に対応している様子である。著作権の問題もあるだろうから，早急に対処すべきだ。私見ではあるが，FM-7にもDOSモードで使用できるF-BASIC（Disk Version）のVersion 2.0の5インチ版あたりを早急にリリースするのが良策か，と考える。

表4　ハードウェア総合評価

	放　熱	動作騒音	ディスケット挿入の容易さ	ケースの堅牢さ	内部塗装他（防錆性）	仕上げ全般
CF-80FM	◎	△	○	◎	○	○
LFD-550/FM	○	○	○	○	△	○
THIN LINE/FM	◎	△	○	◎	◎	○
TF-20	○	◎	◎	○	○	○
KD280/FM-8	○	△	○	○	◎	○
disk-80F	◎	○	○	○	△	○

（注）○印は良好，△印はやや劣るを意味する。

5 最後に

データストレージメモリとしてフロッピーディスクは，現在のマイコン界の主流であることは相違ないようである。前述のように5インチFDDも倍トラック化，倍々トラック化が行われ，メモリ容量の増加が引き続き図られている。しかし，メディアとしてのディスケットはまだ不安定で，まさに言葉どおり“Floppy”（ふにゃふにゃ）であるため，取り扱いの不備や外部からの磁気に対してまったく無防備である。現在も引き続いて，さまざまなメモリデバイスが開発されており，たとえばレーザーディスク，超大容量半導体メモリ，バブルメモリなど枚挙にいとまがないほどである。次世代のメモリストレージがどのデバイスに置き換えられるのか，実に興味深いところである。

追　記

冒頭にこの業界の価格競争の激しさを述べたが，本原稿入稿後，新機種，新定価に改良，改正された物が出た。内訳は以下のとおりである。

○アイテム社　disk-80FII W128,000円
〈オプション〉disk　I/O　15,000円
〈オプション〉ケーブル　5,000円
○工人舎 KD-290/FM-8 W148,000円
S　98,000円

以上は新製品であり，下記は新定価である。

○湘南電送機器社　S　98,000円
W128,000円

FLEX® はTSC社の登録商標です。
OS-9™ はマイクロウエア社の登録商標です。
CP/M® はデジタルリサーチ社の登録商標です。

役に立つプログラムを作るための

ソフトウェア設計法入門講座(2)

Computer Science Group　武原　宰

各社から16ビットパソコンが続々発表されている。FMシリーズにも，８ビット・16ビット両用のFM-11が登場し，処理速度やグラフィック機能面で格段の進歩をとげた。今後，ビジネスユースでは16ビットマシンが主力となることはまちがいないと思われる。

ところで，８ビットと16ビットのマイコンは，どう違うのであろうか。

８ビットのマイコンは，８ビットを単位として，記憶し，計算し，処理を行なう。この処理単位が16ビットであるものを，16ビットマイコンと呼ぶのである。つまり，16ビットのマイコンは一度に計算できる桁数が８ビットの倍あることになる。当然，単純に考えれば計算速度が2倍以上になると言えるわけだ。それはそうだろう，８桁の電卓で16桁の数を計算するのと，16桁の電卓で計算するのを考えれば簡単に理解できる。

この，計算速度の向上も大きいが，もうひとつ大きい点がメモリ管理能力の向上にある。

８ビットCPUでは，基本的に64KBまでのメモリしか使うことができなかった。外部にLSIを付加するなり，回路に工夫を凝らすなりすれば，それ以上のメモリを使うこともできるが，速度の点，操作性の点で非常に問題が多くなるのである。ところが16ビットCPUでは何の苦労もなしに，大規模なメモリ（8ビットCPUの4～16倍程度）を管理できるのだ。

さて，それではメモリが広くなるとどのようなメリットがあるのだろうか。

まず，データ領域が広くなり大量のデータを同時にメモリ上に乗せて処理することが可能になる。メモリ空間が狭いとメモリ上にデータ領域が十分とれなくなり，データをディスクとの間でこま切れにしてやりとりし少しずつ処理するため，たいへんな時間を要することになる。一方，データ領域が広くなると，ディスクとのやりとりがなくなることで，一連の処理に要する時間が大幅に短縮される。そのスピードアップ効果は，ＣＰＵ自体の処理速度の向上よりもはるかに大きくなるのである。

データ領域もだが，プログラム領域が広くなる点も見逃がせない。BASICでプログラムを組んでいて，メモリ不足に悩まされた方々も少なくなかったと思われる。これは複雑で高度な機能を持つプログラムを組み立てて実行できるようになったということだ。また，プログラム領域の拡大は，FORTRAN, PASCAL, COBOLなどの高級言語の使用も可能にする。８ビットマシンでもこれらの高級言語の実行は可能であったが，限られたメモリ内では実用的なプログラムを組むことはできなかった。プログラム領域の拡大により，パソコンにおいても高級言語を実用に供することが可能になった。マシンごとの方言ともいえるBASICと異なり，高級言語は世界の共通語である。しかも別々のプログラムを結びつけて使用することがBASICよりも容易にできるため，プログラムの再利用（以前に作成されたプログラムの一部または全部を別のプログラムで利用すること）が可能になるわけだ。プログラムの再利用，市販プログラムパッケージの利用などはビジネスユースのプログラム開発を安く上げるための重要なキーポイントである。

ここでプログラム設計法が大きな意味を持ってくる。つまり，大規模なプログラムを確実につくりあげ，後々までもそのプログラムを活用するためには，そのプログラムに関するしっかりとしたドキュメンテーション（文書化）が必要不可欠となる。16ビットＣＰＵの広大なメモリ空間を自由自在に操るためのコンパスとなるソフトウェア設計法が，どうしても必要になってくるのである。

創刊号でも紹介したHIPO法(ハイポ法)は，設計法の中でも有力で代表的なものである。

前回はソフトウェア設計法と文書化の必要性について説明した後，HIPO法について概略を説明した。HIPO法はIBM社で開発されたプログラム設計法の一つであり，従来のフローチャートに取って代わることを目的とした手法である。

前回の内容と重複するが，もう一度HIPO法の概要と特徴について触れておこう。

I．HIPO法の特徴

プログラム記述の最も一般的な方法としてフローチャートがあるが，HIPO法はそれに対してどのような利点を持っているのだろうか。それは何よりもHIPOという名前に端的にあらわれている。

Hierarchical Input, Process and Outputの頭文字をとってHIPOと命名した。Hierarchicalとは階層構造を意味する。すなわちHIPO法で設計されたプログラムは複数のモジュール（部分的プログラム）の集まりとして構成され，そのモジュールが階層的な構造を持つことを意味するのだ。プログラムのモジュール化や階層構造については前回で詳しく述べているが，重要な概念であるので次節でもう一度触れることにする。次にInput, Process and Outputであるが，これは入力一処理一出力を意味する。あらゆるプログラムは入力データを取り込み，処理し，

出力データを作りだすのであるが，このInput，Process and Outputという言葉には各過程を明確に表現する手法，という意味がこめられている。過程はIPO図というHIPO法独特の図で表現されることになる。

モジュール化，構造化されたプログラムを階層構造図で表現し，入力―処理―出力の流れをIPO図で表現する，というのがHIPO法の基本なのである。

II. 構造化プログラム

プログラムを小さな部分（モジュール）の集まりとして記述し，モジュール間に階層的な従属関係をもたせたプログラムを「構造化されたプログラム」あるいは「モジュール化プログラム」と呼ぶ。具体的には各モジュールは，サブルーチン・プログラムとして作成され，上位にあるモジュールが下位のモジュールをサブルーチンとしてコールする形式をとる。こうすると，大規模で複雑な処理も，上位モジュールではいくつかの下位モジュールをコールするだけの形で記述することになり，コーディングを見ただけでどのような処理がどのような順序で行われるかが一目瞭然である。

1 構造化プログラムの例

プログラムリストを見ていただきたい。これは前回も掲載したスクリーンエディット型表計算のプログラムだが，HIPO法に基づいて設計した「構造化プログラム」である。

MAINと書かれたモジュールを見てほしい。メインプログラムは全プログラム中の最上位モジュールであり，ここを見ればプログラム全体の動きが理解できるようになっている。MAIN以下の行番号140から240まではほとんどがGOSUB文で構成されている。順を追って見てみよう。

まず，行番号140から160まではディメンジョン，色指定などの前準備である。170は“Line Print”というコメントがあり表の枠を表示する，ということがわかる。続いて180は“Data Print”とあり，枠を表示した後，枠内にデータを埋めていく処理だと見当がつく。190は“LINE INPUT”となっており，キーボード入力待ちである。ここでスクリーン上にデータを書き込み，OKならばReturnキーを押して次の200に進む。200には“Data Input”というコメントがあり，ここでスクリーン上のデータを読み込んでいることがわかる。さらに210では縦方向および横方向の合計を計算し，220でそれをスクリーン上に表示している。そして230で再び入力待ちとなり，入力後は200に戻り入力処理を行う。

図1　階層構造図

2 構造化プログラムの利点

以上のことがリストを見るだけで理解できる。これだけの内容がたった11行のメインプログラムに凝縮されているため，リストを一目見てプログラム全体の動きの概略がわかる。これは「構造化されたプログラム」の強力な利点の一つである。構造化されていない普通のプログラムでは全部を読まないかぎり，全体の動きを把握することができない。

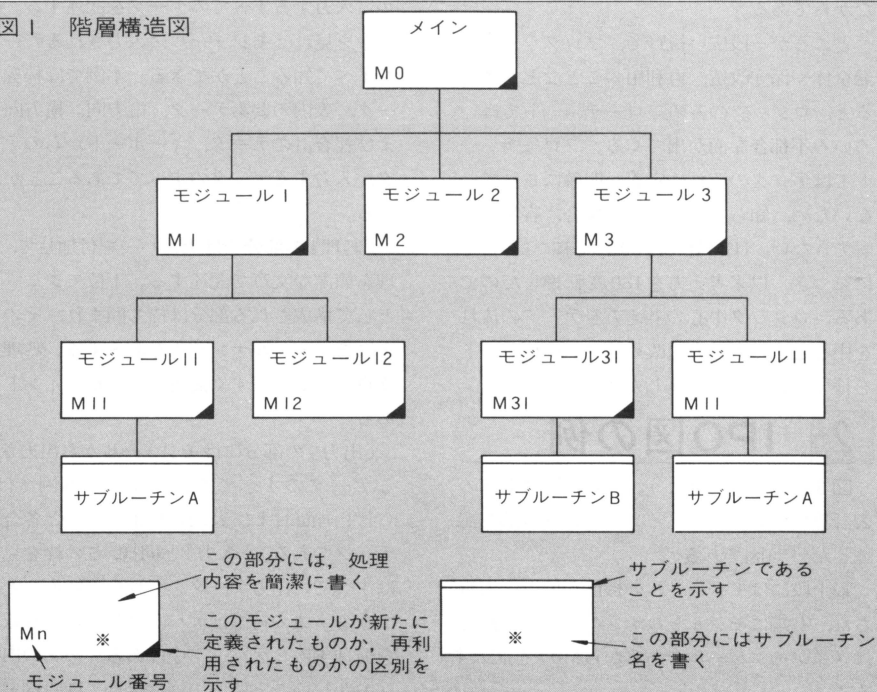
また，階層構造のプログラムは修正や改造がきわめて容易だ，という利点もある。たとえば計算方法を少し変更したいとすれば，行番号3000のCulculateとコメントされたモジュールだけに注目して改造すればよい。

また，入力データをディスクにファイルする，という改造を行う場合には，新たにファイル書き込み用のモジュールを作成し，メインモジュールでそのモジュールをコールすればよい。各処理のどの階層に挿入すればよいかは，メインモジュールの各行を見ながら確実に判断することができる。

3 階層構造図

このような「構造化されたプログラム」の構成をわかりやすく表現したものが，**図1**の階層構造図である。メイン，モジュール，サブルーチンなどの用語やモジュール番号の付け方は前回説明したので省略する。**図2**は例として用いたスクリーンエディット型表計算プログラムの階層構造図である。

大規模なプログラムになると階層数が十数段にもなり，モジュールの数も何百個にもなるが，モジュール1個当たりの大きさは，リストにして数行から数十行以内に納められる。デバッグや改造時には特定の限られたモジュールにだけ着目すればよい。リストの洪水に溺れなくとも済むのである。

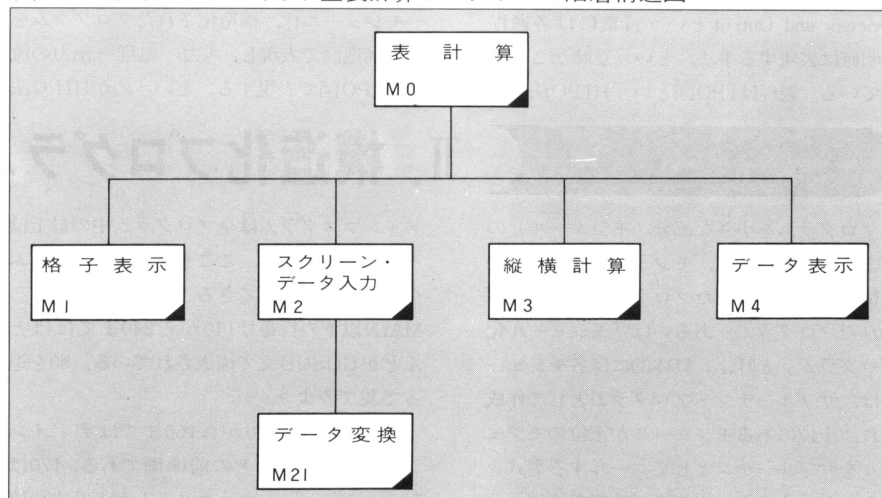
このようにプログラムが大きくなればなるほど「構造化されたプログラム」は威力を発揮する。

4 モジュール分割

理解しやすいプログラム，すなわちデバッグが容易で(したがって信頼性が高い)，かつ改造や他のプログラムへの転用も可能なプログラムを作るには，プログラムのモジュール分割がたいせつなキーポイントとなる。モジュール化の利点をいろいろ述べたが，それらの利点を十分に発揮させるには，うまい具合にモジュールを切り分ける必要がある。下手に切り分けるとモジュール同士のつながりが強くなりすぎて，一つのモジュールを改造するとそれに関連した他のモジュールまで改造しなければならなくなるからだ。HIPO法では最適なモジュール分割に導くための手法もちゃんと用意されているが，専門的になりすぎるので，また次の機会に譲ろうと思う。

ここではモジュール分割を行うときに注意すべき点を簡単にまとめておこう。

①モジュール1個当たりの機能は，文章1行で表現できる程度に単純化する。

②モジュール1個当たりの行数は20行以内とする。

③上位モジュール，特にメインモジュールは原則としてサブルーチンコール(GOSUB文)のみで構成するように心がける。

④他のモジュールとの間で受け渡すデータの数をできるかぎり少なくする。

以上である。このうち④はモジュールの独立性を保つという意味から，特に重要な項目である。

図2　スクリーンエディット型表計算プログラムの階層構造図

プログラム実行例

25	0	-5	20
0	0	0	0
0	80	-100	-20
0	0	0	0
0	0	0	0
-80	-56	400	264
-55	24	295	264

Ⅲ. 入力—処理—出力の表現

HIPO法でフローチャートのかわりに書くのがIPO図である。Input(入力)，Process(処理)，Output(出力)の頭文字を取ってIPO図である。

1 フローチャートとの違い

フローチャートがロジックの流れを中心に記述するのに対し，IPO図では入力—処理—出力というデータの流れに着目してプログラムを記述する。ロジックの流れに着目するか，データの流れに着目するかというのがフローチャートとIPO図の根本的な違いである。

通常プログラムを設計しているとき，設計者の頭の中はロジックの流れを追跡している。データの流れはあらかじめ別に考えておいて，それに基づいてロジックを考えていくわけだが，プログラムそのものを追っていく場合には，ロジックの流れに着目してフローチャートで記述する方がやりやすいからである。このことはコンピュータが誕生して以来，20年以上にわたり，プログラムの記述はロジックを追跡したフローチャートのみであった事実によっても証明される。プログラムの作成だけを考えるとそれがいちばんやりやすい方法だからである。

ところが一段広い視野で，プログラムの開発全体や保守改造，再利用のことなどを考えると，ロジックのみのフローチャートではいろいろ不都合な面が出てくる。フローチャートではデータの流れや定義が明確にされていないため，知らない人が見た場合，容易に理解できない。IBM社では60年代末にこのことに気づき，以来考え方を180度転換したのである。ロジック中心の考え方をデータの流れを中心とする考え方に改め，フローチャートを捨ててIPO図を考案したのだ。

2 IPO図の例

図3を見ていただこう。これはリスト例の表計算プログラムのメインモジュール部を記述したIPO図である。

最上段にはモジュールの機能が簡潔に表現され，右端にモジュールナンバー（この例ではメインモジュールであるからM0）を記入する。

その下に大きな枠が3個配置されており，〔入力〕，〔処理〕，〔出力〕とタイトルがつけられている。〔入力〕の枠の中にはこのモジュールに入力するすべてのデータを記述する。枠の中を見ればモジュールに入力されるデータをすべて知ることができる。本例では枠線データ，表内の要素データ，縦方向，横方向および総合計のデータ，キーボードからのデータが入力するデータのすべてであることがわかる。

〔処理〕の部分ではナンバーを付加して，処理を簡潔な文章で記述する。下位モジュールとして構成される部分は枠で囲まれ，その右下にモジュールナンバーを記入する。処理を文章でわかりやすく記述するのがポイントである。

〔出力〕の部分にはモジュールから出力されるデータのすべてを記入する。本例では，CRT画面および表内の画素データと各合計データである。〔入力〕と同様，この枠を見ればすべての出力データをもれなく知ることができる。この特徴はデバッグ時に非常な威力を発揮する。デバッグ中にあるデータの内容がおかしいことに気づいた場合，IPO図を取り出し〔出力〕欄をながめれば，問題となって

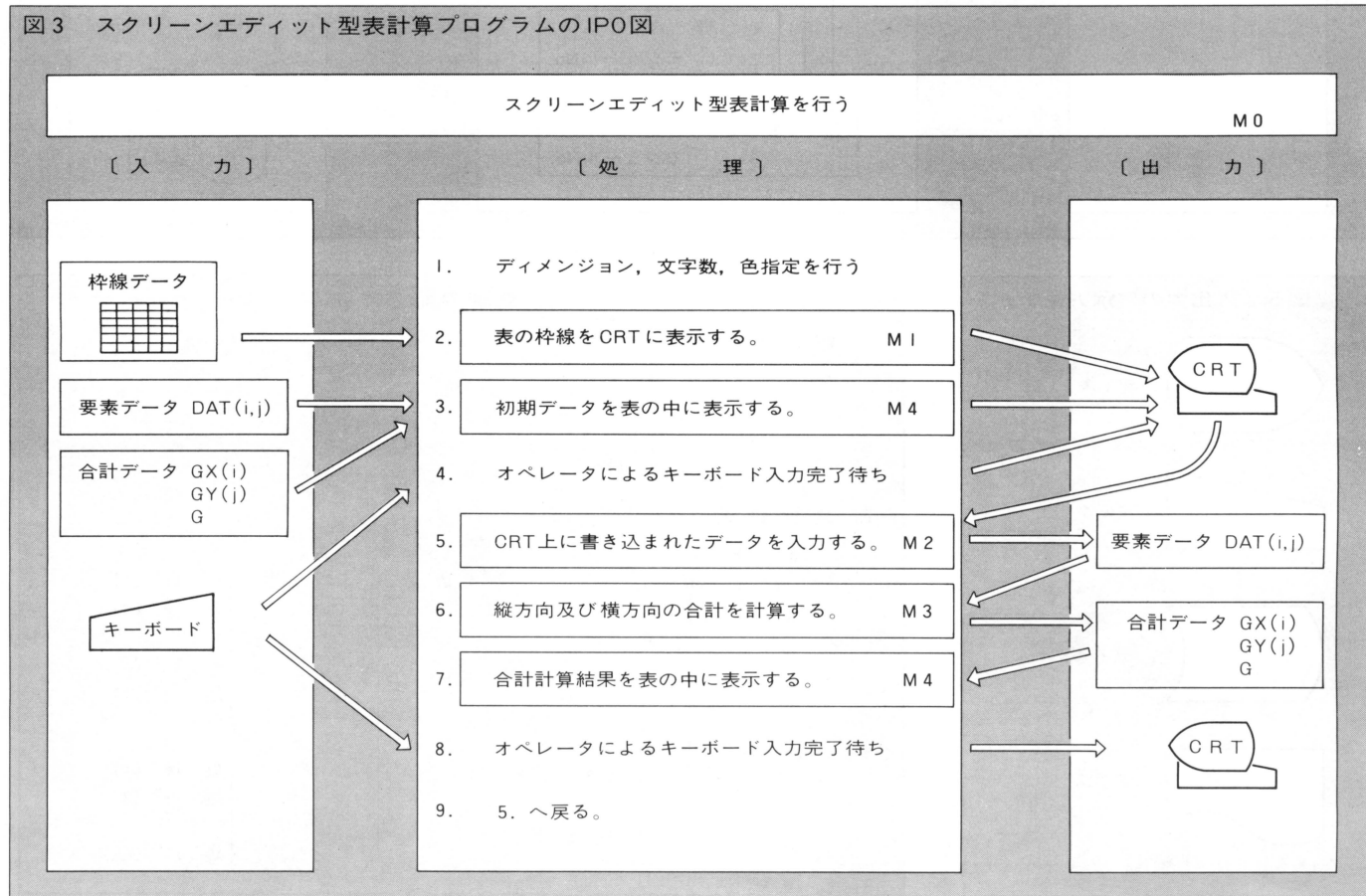
いるデータの出力しているモジュールと処理が，ただちにわかる。

さて，〔入力〕と〔処理〕，〔出力〕間は矢印で結ばれている。これはデータの流れを表しており，〔入力〕から〔処理〕へ，〔処理〕から〔出力〕へと流れている。ただ例外的に，出力データの一部から処理に流れる矢印があるが，処理結果を再び次の処理へ入力することを示している。

順を追って見てみよう。

まず処理の1で初期設定を行う。次に2で表の枠を表示するが，これはモジュール1で行われる。この処理への入力は枠線データ，出力はCRT上に行われるとわかる。処理3はモジュール4により，初期データを表内に表示する。初期データは要素データDAT(i, j)および合計データGX(i)，GY(j)，Gより成り，これをCRT上に出力する。処理4ではオペレータによりキーボードから入力された値がCRT上に表示され，処理5でそのデータをCRTから入力する。入力はモジュールで行われ，結果は要素データDAT(i, j)として出力される。処理6ではその要素データを入力して，縦方向および横方向の合計を計算する。計算はモジュール3で行われ，結果はGX(i)，GY(j)，Gとして出力される。処理7ではその合計データを表内に表示する。

以上のようにIPO図を見れば，処理とそれに伴うデータの流れが把握できる。従来のフローチャートは，IPO図の処理の部分のみを記述したものであり，データ部は別に一覧表を用意したが，IPO図ではこれらを1枚の図に表現できる。

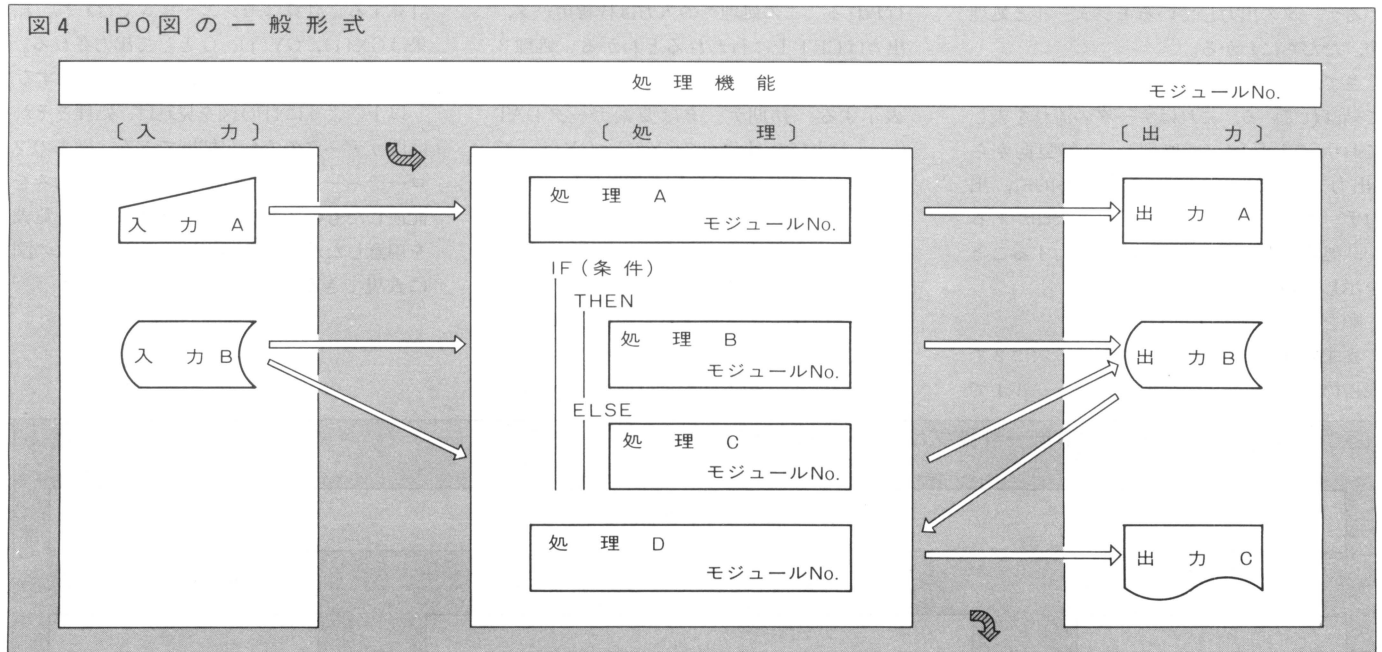
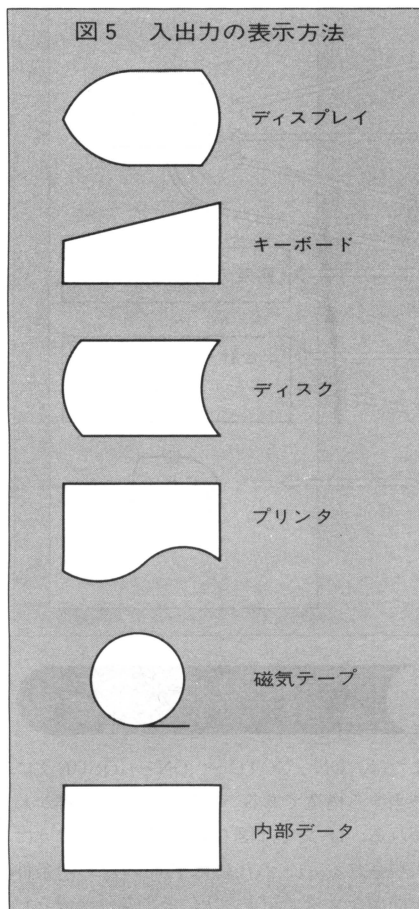
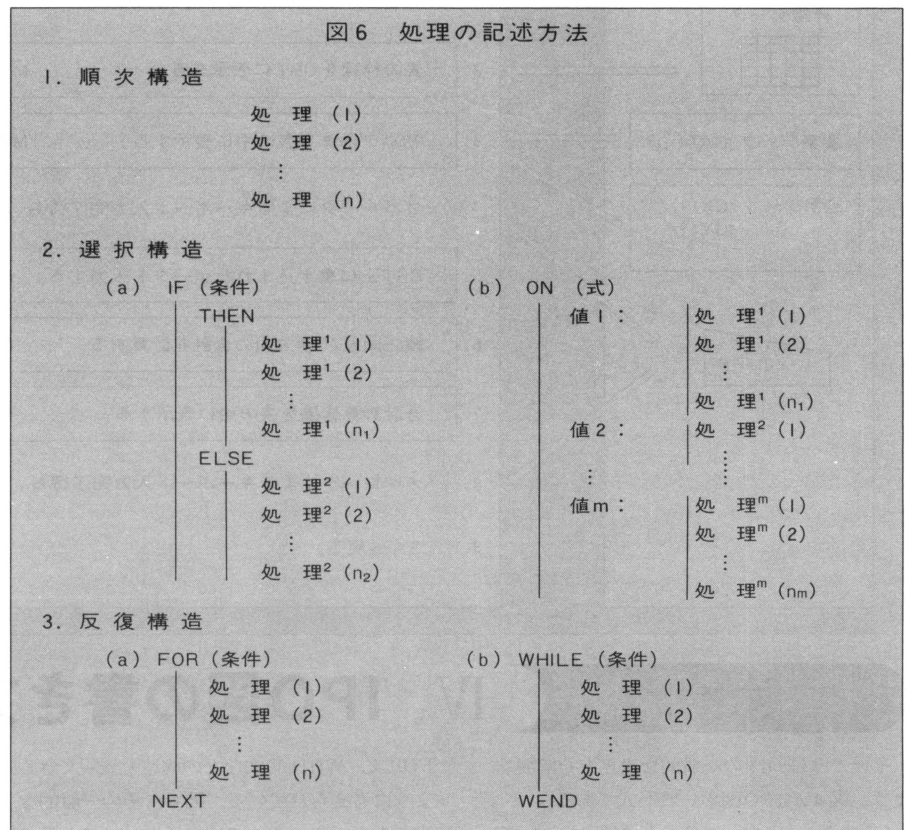
図3　スクリーンエディット型表計算プログラムのIPO図

Ⅳ．IPO図の書き方

それではIPO図の一般的な書き方を解説しよう。図4がIPO図の一般形式である。

〔入力〕と〔出力〕の部分は形態に応じて図5の図式が用いられる。さらに詳しいデータの構造を示す必要がある場合には，枠の中に図式的に表現する。記述はモジュールのレベルに応じた必要十分なものとし，不必要に詳しい記述は避ける。

次に〔処理〕の部分である。図3の例では順番に処理を並べただけであるが，IF文やFOR文，WHILE文などの構造も表現できるように考えられている。図6にその一般的な書き方を示す。ここではマイクロソフト系のBASICに含まれるもののみを選んだ。

処理記述の方法は，順次構造，選択構造，反復構造の三つに分けられる。順次構造とはその中に分岐や繰り返しを含まず，順番に一連の処理を行う構造であり，図3の処理はこれに当たる。

選択構造はBASICのIF～THEN～ELSEまたは，ON～GOTO～，ON～GOSUB文に該当する構造である。条件によって処理がわかれる。IF文では図6・2(a)に示すように表現される。IFの後には判断のための条件を簡潔な文章で記す。次にその条件が成立した場合に実行する処理を，THEN以下に順次並べる。成立しない場合の処理はELSE以下に並べる。そして最後に，IF文，THEN，ELSEの範囲を示すための縦線を，おのおのに引く。

ON～GOTO～文，ON～GOSUB～文は図6・2(b)のように記述する。ON文は判断基準となる式の値によって処理が分岐する。各値に応じた処理を図のように，値の後ろに並べる。ON文と各値の範囲を明示するための縦線を引くことは，IF文の場合と同様である。

反復構造はBASICのFOR文とWHILE文に対応する。図6・3(a)にFOR文，(b)にWHILE文の記述方法を示す。両方とも（条件）と書かれた部分に，繰り返しを続ける条件を，簡潔な文章で記述する。この場合も他と同じように繰り返すべき処理を並べ，縦線で繰り返し範囲で明示する。

各処理の中がさらに選択構造や反復構造となる場合もあるが，複雑になるときにはそれらの一連の処理を，新たな下位モジュールとしてまとめ，できるかぎり一段構造とするのが望ましい。

```
10   '■──────────────────────────────────────────────────────■
15   '■                                                      ■
20   '■       SCREEN    EDIT    CULCULATION    TABLE         ■
25   '■                                                      ■
30   '■                                                      ■
35   '■                          Copyright by                ■
40   '■                                                      ■
45   '■                               Computer Sciense       ■
50   '■                                                      ■
55   '■                               1982 08/17             ■
60   '■──────────────────────────────────────────────────────■
100  '■──────────────────■
110  '■     MAIN         ■
120  '■           M.0    ■
130  '■──────────────────■
140 DIM DAT(6,3):DIM GX(6):DIM GY(3):DIM A%(4)
150 CLS:WIDTH 40,20
160 COLOR 5
170 GOSUB 1000:'*** Line Print(M.1) ***'
180 GOSUB 4000:'*** Data Print(M.4) ***'
190 LINE INPUT Y$
200 GOSUB 2000:'*** Data Input(M.2) ***'
210 GOSUB 3000:'*** Culculate X,Y(M.3) ***'
220 GOSUB 4000:'*** Data Print(M.4) ***'
230 LINE INPUT Y$
240 GOTO 200
1000 '■──────────────────────■
1010 '■  Line   Print        ■
1020 '■                M.1   ■
1030 '■──────────────────────■
1040 LOCATE 0,3
1050 PRINT"     ┌──────┬──────┬──────┬┬──────┐"
1060 PRINT"     │      │      │      ││      │"
1070 PRINT"     ├──────┼──────┼──────┼┼──────┤"
1080 PRINT"     │      │      │      ││      │"
1090 PRINT"     ├──────┼──────┼──────┼┼──────┤"
1100 PRINT"     │      │      │      ││      │"
1110 PRINT"     ├──────┼──────┼──────┼┼──────┤"
1120 PRINT"     │      │      │      ││      │"
1130 PRINT"     ├──────┼──────┼──────┼┼──────┤"
1140 PRINT"     │      │      │      ││      │"
1150 PRINT"     ├──────┼──────┼──────┼┼──────┤"
1160 PRINT"     │      │      │      ││      │"
1170 PRINT"     ╞══════╪══════╪══════╪╪══════╡"
1180 PRINT"     │      │      │      ││      │"
1190 PRINT"     └──────┴──────┴──────┴┴──────┘"
1200 RETURN
2000 '■──────────────────■
2010 '■ Data Input       ■
2015 '■           M.2    ■
2020 '■──────────────────■
2030 FOR I=1 TO 6
2040 FOR J=1 TO 3
2050 X1=(J-1)*8+4:Y1=(I-1)*2+4
2060 Y2=Y1
2070 X2=X1+7
2080 GET@ (X1,Y1)-(X2,Y2),A%
2090 GOSUB 21000:'*** DATA CONVERT(M.2.1) ***'
2100 NEXT J,I
2110 RETURN
```

```
3000 '■―――――――――――――――――――――――――――――――――■
3030 '■ Culculate GX(I),GY(J),G        ■
3060 '■                          M.3 ■
3090 '■_________________________________■
3120 FOR J=1 TO 3
3150 GY(J)=0
3180 FOR I=1 TO 6
3210 GY(J)=DAT(I,J)+GY(J)
3240 NEXT I,J
3270 FOR I=1 TO 6
3300 GX(I)=0
3330 FOR J=1 TO 3
3360 GX(I)=DAT(I,J)+GX(I)
3390 NEXT J,I
3420 G=0
3450 FOR J=1 TO 3
3480 G=GY(J)+G
3510 NEXT J
3540 RETURN
4000 '■―――――――――――――――――――――■
4010 '■ Data Display         ■
4020 '■                 M.4 ■
4030 '■_____________________■
4040 LOCATE 0,0:COLOR 4
4050 FOR I=1 TO 6
4060 FOR J=1 TO 3
4070 LOCATE (J-1)*8+6,(I-1)*2+4
4080 IF DAT(I,J)<0 THEN COLOR 2 ELSE COLOR 4
4090 PRINT USING"######";DAT(I,J);:PRINT" "
4100 COLOR 5:LOCATE (J-1)*8+6+7,(I-1)*2+4:PRINT"|"
4110 NEXT J,I
4120 FOR J=1 TO 3
4130 LOCATE(J-1)*8+6,16
4140 IF GY(J)<0 THEN COLOR 2 ELSE COLOR 4
4150 PRINT USING"######";GY(J)
4160 NEXT J
4170 FOR I=1 TO 6
4180 LOCATE 31,(I-1)*2+4
4190 IF GX(I)<0 THEN COLOR 2 ELSE COLOR 4
4200 PRINT USING"######";GX(I)
4210 NEXT I
4220 LOCATE 31,16
4230 IF G<0 THEN COLOR 2 ELSE COLOR 4
4240 PRINT USING"######";G
4245 LOCATE 0,0
4250 RETURN
21000 '■―――――――――――――――――――――――――――――■
21010 '■  DATA CONVERT               ■
21020 '■                    M.21     ■
21030 '■_____________________________■
21040 SMARK=0
21050 FOR K=1 TO 3
21060 IF (A%(K) AND &HOFOO)=&HODOO THEN A%(K)=(A%(K) AND &HOOFF):SMARK=1
21070 IF (A%(K) AND &HOOOF)=&HOOOD THEN A%(K)=(A%(K) AND &HOOOO):SMARK=1
21080 NEXT K
21090 DAT(I,J)=((A%(1) AND &HOFOO)/256)*100000!+(A%(1) AND &HOOOF)*10000!+((A%(2
) AND &HOFOO)/256)*1000!+(A%(2) AND &HOOOF)*100!+((A%(3) AND &HOFOO)/256)*10!+(A
%(3) AND &HOOOF)
21100 IF SMARK=1 THEN DAT(I,J)=-DAT(I,J)
21110 RETURN
```

for FM-7, -8, -11

6809 マシン語入門（後）

鶴岡哲明

さて，マシン語入門の後編であるが，まず前編についての訂正と補足を述べさせていただきたい。

プログラム5についてであるが，2行目のＳＴＡの次の＃は，もちろん付けてはいけない。そして4行目のコメント「～＄7000に加えていく」というのは「＄7001に加えていく」の誤りである。10行目のＢＥＱはＢＮＥの誤りである。以上，訂正してお詫びする。

次に補足であるが，このマシン語はFM-8ということで話を進めてきたが，FM-7やFM-11でもマシン語は同一である。ただハードウェア構成の違いや，BASICの違いで自由に使ってよいメモリエリアが，多少異なるので，プログラムを打ち込む際，場合によっては多少の変更が必要となる。つまり前号のプログラムでは$7000番地から$71FF番地を使ったが，FM-7, FM-8のディスクシステムの場合は，$6000番地に移すなどの作業が必要である。今回はすべてのＦＭシリーズで有効になるように，$6000番地あたりを使っていく。このとき，

CLEAR 300＆H5FFF

を行っておいてほしい。

なお，図表番号は前号と通しで付けてあるので，その点注意していただきたい。

16 実行速度

まず命令の実行速度について触れたい。命令の実行には，命令表に記載されているサイクル数だけ時間がかかる。また，次章で述べるインデックスアドレッシングなどの，ポストバイトのつくものはポストバイト表と照らし合わせてサイクル数を計算しなければならない。1サイクルは，CPUのクロックの1サイクルなので，FM-8の場合1/1230000秒，FM-7，11は1/2000000秒となる。80系と比べると，68系CPUのクロック周波数は低いが，これはクロックの使い方が異なるためで，単純な比較はできないが6809の2MHzなら，Z80の4MHzより速いと思う。

17 インデックスアドレッシング

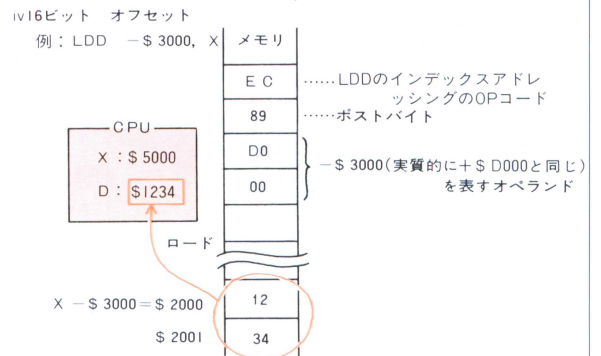
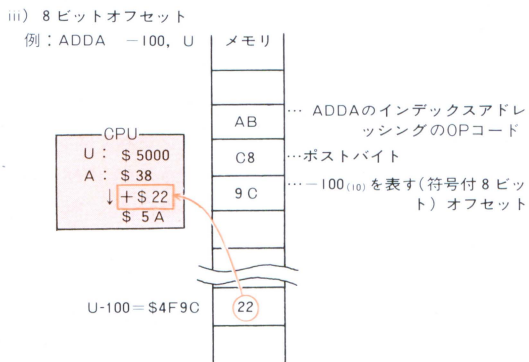
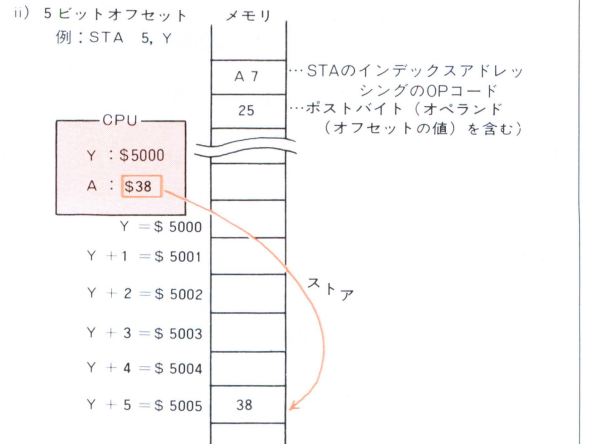
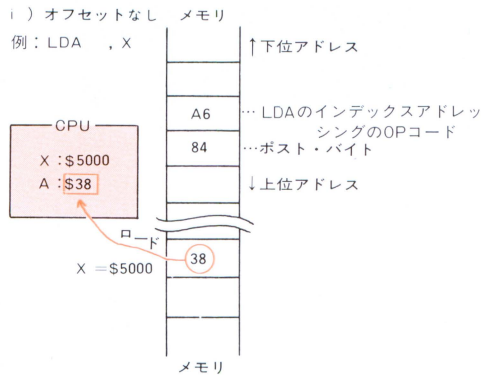
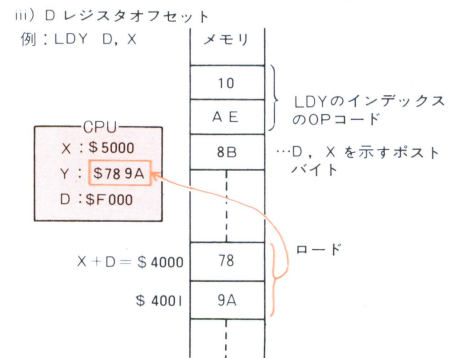
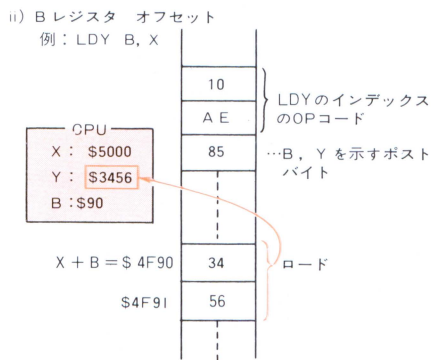
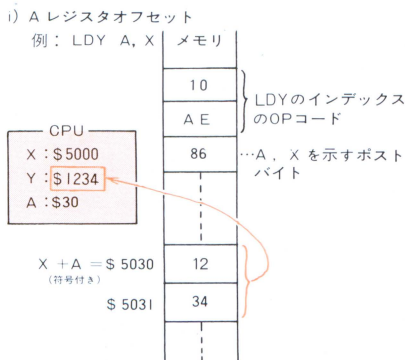
6809の最強力なアドレッシングモードであるインデックスアドレッシングを紹介しよう。これはレジスタX，Y，U，Sの値により参照するアドレスが変えられるアドレッシングモードで，さらにバリエーションがあるが，まず**表4**の①～④を理解していただきたい。**図5**のa)にその参照の様子を示した。

実効アドレスというのはアドレッシングモードによって指定されるこれから参照するアドレスのことで，イミディエイトアドレッシングではオペレーションコード（略してOPコード）の次の数値で，エクステンドアドレッシングではOPコードの次の2バイトで定数で与えた。インデックスアドレッシングでは，レジスタの値から決められた手順で，実効アドレスを計算するわけだが，レジスタの値を変えることで実効アドレスが変えられ，大量のデ

表4　インデックスアドレッシングの種類とポストバイト

		実効アドレス	
①オフセットなし		R	
②5ビットオフセット	n, R（ただし－16≦n≦15）	R＋n	
③8ビットオフセット	n, R（ただし－128≦n≦127）	R＋n	
④16ビットオフセット	n, R（ただし－32768≦n≦32767）	R＋n	
⑤Aレジスタオフセット	A, R（Aは符号付きの数として加算）	R＋A	
⑥Bレジスタオフセット	B, R（Bは符号付きの数として加算）	R＋B	
⑦Dレジスタオフセット	D, R（Dは符号付きの数として加算）	R＋D	
⑧オフセットなし・ポストインクリメント＋1	, R＋	R	①の実行後LEAR 1, Rを実行するのと等価
⑨オフセットなし・ポストインクリメント＋2	, R＋＋	R	①の実行後LEAR 2, Rを実行するのと等価
⑩オフセットなし・プリデクリメント－1	, －R	R－1	①の実行前にLEAR －1, Rを実行するのと等価
⑪オフセットなし・プリデクリメント－2	, －－R	R－2	①の実行前にLEAR －2, Rを実行するのと等価
⑫PCリラティブ8ビットオフセット	n, PCR	PC＋n	
⑬PCリラティブ16ビットオフセット	n, PCR	PC＋n	

＊この表におけるn（インデックスレジスタからの変位）をオフセットという。

図5　インデックスアドレッシング
（図中のCPU内のレジスタの値は一つの例である）

ータを扱うときに欠かせない。たとえば，

LDA　,X

とすると，Xレジスタの値がそのまま実効アドレスになる。

さて，ここで問題なのはコード化の作業（コーディング）である。**表4**のとおりインデックスアドレッシングには種類が多くあり，とても1バイトのOPコードでは表しきれない。そこで，ポストバイトというものをつける。インデックスアドレッシングをアセンブルする場合，まず命令表で目的の命令のインデックスアドレッシングのコードを調べ，次にインデックスアドレッシングのポストバイト表（前号**別表2**もしくは今号**別表6**）を見てポストバイトのコードを求める。その後は必要に応じてオフセットを表す部分をつければよい。ポストバイトは**別表2**では2進数で表記してあるので，アセンブラを持たない人は，これを16進数に書き直さなければならない。この手間を少しでも軽くするため16進表記したものを**別表6**に示す。

なお特に注意が必要なのは，**表4**の②の5ビットオフセットの場合のポストバイトである。この場合，5ビットオフセットは符号付き（signed）で表現する。つまり，

$-16_{(10)} \rightarrow 10000_{(2)}$
$-1_{(10)} \rightarrow 11111_{(2)}$
$5_{(10)} \rightarrow 00101_{(2)}$

というようになる。これを**別表2**のnnnnnにはめこめばいいわけだ。**別表6**を用いる場合にはオフセットnの符号によって表中の指定のようにNを求め，これを下位4ビット（16進1桁）とすればよい。表4の①はただポストバイトをつけ，③はポストバイトの後に8ビット符号付きで表現したオフセットをつける。④も同様に16ビット符号付きで表現した

〔別表６〕 MC6809 POST BYTE

(The superscript/subscript pair printed beside each code is given as ±/‡: ± = added machine cycles, ‡ = added bytes.)

インデックス ポストバイト			非インダイレクト X	±/‡	Y	±/‡	U	±/‡	S	±/‡	PC(1)	±/‡	インダイレクト X	±/‡	Y	±/‡	U	±/‡	S	±/‡	PC(2)	±/‡	インデックスなし	±/‡
コンスタントオフセット	ゼロオフセット ,R		84	0/0	A4	0/0	C4	0/0	E4	0/0	—		94	3/0	B4	3/0	D4	3/0	F4	3/0	—		—	
	5ビット n,R	n＝N	0N	1/0	2N	1/0	4N	1/0	6N	1/0	—		—		—		—		—		—		—	
		n＝N−＄10	1N	1/0	3N	1/0	5N	1/0	7N	1/0	—		—		—		—		—		—		—	
	8ビットオフセット n,R		88	1/1	A8	1/1	C8	1/1	E8	1/1	CC	1/1	98	4/1	B8	4/1	D8	4/1	F8	4/1	DC	4/1	—	
	16ビットオフセット n,R		89	4/2	A9	4/2	C9	4/2	E9	4/2	CD	5/2	99	7/2	B9	7/2	D9	7/2	F9	7/2	DD	8/2	9F	5/2
アキュムレータオフセット	Aレジスタオフセット A,R		86	1/0	A6	1/0	C6	1/0	E6	1/0	—		96	4/0	B6	4/0	D6	4/0	F6	4/0	—		—	
	Bレジスタオフセット B,R		85	1/0	A5	1/0	C5	1/0	E5	1/0	—		95	4/0	B5	4/0	D5	4/0	F5	4/0	—		—	
	Dレジスタオフセット D,R		8B	4/0	AB	4/0	CB	4/0	EB	4/0	—		9B	7/0	BB	7/0	DB	7/0	FB	7/0	—		—	
自動 INC・DEC	インクリメント(＋1) ,R＋		80	2/0	A0	2/0	C0	2/0	E0	2/0	—		—		—		—		—		—		—	
	インクリメント(＋2) ,R＋＋		81	3/0	A1	3/0	C1	3/0	E1	3/0	—		91	6/0	B1	6/0	D1	6/0	F1	6/0	—		—	
	デクリメント(−1) ,−R		82	2/0	A2	2/0	C2	2/0	E2	2/0	—		—		—		—		—		—		—	
	デクリメント(−2) ,−−R		83	3/0	A3	3/0	C3	3/0	E3	3/0	—		93	6/0	B3	6/0	D3	6/0	F3	6/0	—		—	

(1) 上位のCは8,A,Eも可　(2) 上位のDは9,B,Fも可　±：追加されるマシンサイクル数　‡：追加されるバイト数

オフセットを，上位，下位バイトの順でつける。ただしこの場合，オフセットもそれを加えるレジスタも16ビットなので，16ビットを越える繰り上がり，繰り下がりはアドレス計算時には無視される。よって，このオフセットは符号なし（unsigned）と考えてもさしつかえない。

以上が定数オフセットの場合であるが，アキュムレータオフセットでもまったく同様である。表4の⑤，⑥はAレジスタ，Bレジスタを8ビット符号付きオフセットとするもので，③の場合，命令中のデータをオフセットに使うのに対し，アキュムレータの値をオフセットに使う。Dレジスタオフセットも同じで，④の16ビットオフセットのかわりにDレジスタを使う。

アキュムレータオフセットは変えられるものを2つ持つわけで，非常に便利なアドレッシングモードである。たとえばあるアドレスからデータがいくつか並んでいて，調べたいデータが何番めにあるかがBレジスタに入っていて，データの先頭アドレスがXレジスタに入っていると，

　　LDA　B，X

として，そのデータを読み出せる。

ある程度マシン語に慣れてくると，このありがたみがよくわかる。80系統のCPUではこう単純にはいかない。

18 LEA命令

インデックスアドレッシングにはまだ種類があるが，6809の特徴として前記のようなアドレス計算をするアドレッシングモードを持つばかりでなく，インデックスアドレッシングで指定されるアドレスの値をレジスタに格納する命令がある。それは LEA （Load Effective Address）命令である。実効アドレスをレジスタに入れるのだが，この命令はアドレス計算用のものなので，X，Y，U，Sの4つのレジスタが対象であり，ニモニックは，LEAX，LEAY，LEAU，LEASとなる。アドレッシングモードもインデックスアドレッシングしかない。なぜなら，エクステンドアドレッシングの場合はロード命令でできるし，イミディエイトの実効アドレスを計算するなどまず不要だからである。例をあげると，

　　LEAX　1，X

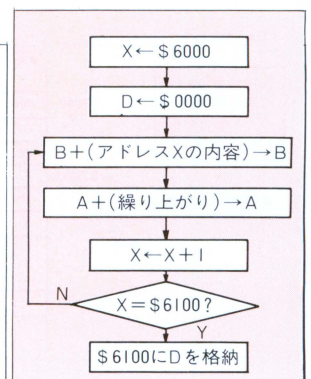
とすれば，（Xレジスタの値）＋1が実効アドレスとなるので，XレジスタにXレジスタの値＋1を代入するので，これによってXレジスタの値を変えられるのである。

つまりLEA命令で，インデックスレジスタに対する加算ができるわけだ。また，前述のアキュムレータアドレッシングを用いれば，インデックスレジスタとアキュムレータの加算ができて便利なことが多い。ただしこの命令はあくまでアドレスを求めるもので，フラグの変化はほとんどない。X，Yレジスタの場合だけは，レジスタをカウンタなどとして使うとき楽なように，Zフラグだけが変化する。

これでインデックスレジスタの変更法がわかったと思うので，例として＄6000番地から＄60FF番地の1バイトのデータ256個を足し，結果を2バイトで＄6100番地に入れるプログラムをあげた（**プログラム6**）。これは結果の上位バイトは，256個の数値の平均の整数部分ともいえる。データの数は設定値の変更だけでよいので，大量のデータを処理させることによってBASICより格段スピードが速い，ということがわかるだろう。ほんとうはVRAMを扱う例をあげたいが，FMシリーズはサブCPUの管理下にあるので，次の機会に譲りたい。

LEA命令とLD命令をうまく使えば，データのあるアドレスの値（ポインタとしてのデータ）をメモリに置くとき，非常に便利である。

プログラム６

```
addr code
6200 8E    6000         LDX   #$6000
6203 CC    0000         LDD   #$0000
6206 EB    84    LOOP   ADDB  ,X
6208 89    00           ADCA  #$00
620A 30    01           LEAX  1,X
620C 8C    6100         CMPX  #$6100
620F 26    F5           BNE   LOOP
6211 FD    6100         STD   $6100
6214 39                 RTS
```

19 ポストインクリメントとプリデクリメント

さて，もう一度**プログラム 6** を見てほしい。

このようにインデックスレジスタを１つずつ増やす（または減らす）場合は比較的多い。そこで，インデックスアドレッシングでメモリを参照するとき，レジスタのインクリメント（増やす）・デクリメント（減らす）を行ってしまうアドレッシングモードがある。これが，インデックスアドレッシング・ポスト（後）インクリメント（**表 4** の⑧，⑨），インデックスアドレッシング・プリ（前）デクリメント（**表 4** の⑩，⑪）だ。

まず，ポストインクリメントから説明する。**表 4** の⑧にある＋１の方は，所定の動作終了後，インデックス用に使ったレジスタを１増やすものである。これを使えば**プログラム 6** は**プログラム 7** のようにシンプルになる。

プログラム 7

```
addr code
6200 8E    6000        LDX   #$6000
6203 CC    0000        LDD   #$0000
6206 EB    80    LOOP  ADDB  ,X+
6208 89    00          ADCA  #$00
620A 8C    6100        CMPX  #$6100
620D 26    F7          BNE   LOOP
620F FD    6100        STD   $6100
6212 39                RTS
```

また6809の命令は16ビットまとめて処理する場合も多い。このとき＋１は不適切で，＋２にするのが普通である。この処理をするのが**表 4** の⑨で，**プログラム 8** にこの例をあげる。これは＄6000に入れられたアドレスから＄6002に入れられたアドレスまでのメモリをクリアするプログラムである。モニタのMコマンドを使ってこの値を決め，実行すればいい。このときむやみやたらなアドレスを設定するとワークエリアを壊し，暴走のもととなるので空きエリアを設定するようにしてほしい（BASICのCLEAR命令を用いて設定するなど）。

プログラム 8

```
addr code
6100 BE    6000        LDX   $6000
6103 CC    0000        LDD   #$0000
6106 ED    81    LOOP  STD   ,X++
6108 BC    6002        CMPX  $6002
610B 23    F9          BLS   LOOP
610D 39                RTS
```

図 6

プリデクリメントについても同様で**表 4** の⑩，⑪を見てもらえばわかるであろう。これはアドレスの大きい方から順に処理するときに使う。ただ注意してもらいたいのは，この場合，命令の処理の前にレジスタをデクリメントする点である。これは後述するスタックとの共通性をもたせるためであるが，端的にいうと⑧のあとに⑪を使うと参照するアドレスは同じになる。これについては**図 6** を見てもらえばすぐわかると思うが，詳しくはスタックの説明に譲る。

20 プログラムカウンタリラティブ

6809はプログラムカウンタもインデックスレジスタとして使える。定数オフセットしかないが，８ビットと16ビットの２とおりある。実効アドレスの計算はインデックスアドレッシングと同じであるが，計算に使われるプログラムカウンタは，その命令実行中にすでに次の命令のあるアドレスの値となっているところに注意してほしい。これは条件分岐のリラティブアドレッシングの飛び先に相当するアドレスが実効アドレスとなる。データ参照の様子は**図 7** を見てほしい。このアドレッシングモードは，プログラムを特定のアドレスと関係なく動作させる場合に有効である。

たとえば定数のデータ・テーブル（表，目録といった意味），BASICでいえばDATA文中に記すデータは，できればプログラムと一緒にくっついている方がよいが，プログラムを他のアドレスに移した場合，データのアドレスもプログラムとともに変わってしまう。その際，このアドレッシングモードを使えばよい。

しかしハンドアセンブル（手作業でするアセンブル）をする場合は，オフセットの計算が面倒である。

また，プログラムカウンタの値は命令を実行すると変わってしまうので，LEA命令などを使って，いったんレジスタやメモリに格納する方がいいだろう。

たとえば，

LEAX　0，PC

とすれば，この命令の次にくる命令のアドレスの値がXレジスタに入るわけである。また，後述するTFR命令を使うとこのアドレッシングモードを使わなくても済む。

図 7　プログラムカウンタリラティブ

$1000

21 インダイレクトアドレッシング

さらに6809にはインダイレクトアドレッシングというものが使える。これには，インデックストインダイレクトとエクステンディドインダイレクトの２つがある。

インダイレクト（間接）とつくのは，今まで紹介してきたアドレッシングモードで指定されたアドレスが目的のアドレスではなく，そのアドレスに入っている２バイトのデータが目的のアドレスとなるからだ。１回のメモリ参照で目的のデータにたどりつけないから，まさに間接である。

まず，インデックストインダイレクトの方から説明する。これは先述したインデックスアドレッシング（プログラムカウンタリラティブを含む）で示される実効アドレスに入っているデータ２バイトが，実効アドレスになる。インデックスでインダイレクト（目的のデータの入っているアドレスを示すデータ）のデータを求める，ということだ。

たとえば，

```
LDA 〔5，X〕
```

を，Ｙレジスタを使ってインダイレクトなしで済ませれば，

```
LDY 5，X
LDA ，Y
```

と同じ結果がＡレジスタに入る。

インデックストインダイレクトは，その機構上，ポストインクリメントやプリデクリメントの＋１や－１は使えない。なぜならインデックスでまず実効アドレスを見るわけで，これは当然，２バイトだから＋１，－１はまずいわけだ。また，５ビットオフセットは，ポストバイト１バイトではもう空いているところがないので存在しない。

エクステンディドインダイレクトも同様で，エクステンドアドレッシングで示されるアドレスに入っている２バイトのデータが実効アドレスとなる。インダイレクトアドレッシングの例を**図８**にあげておいたので参考にしてほしい。

なお，インダイレクトの表記は，すでに例であげたように，インデックスやエクステンドのオペランドを〔 〕で囲んで表記する。

そしてコードにするときは，インデックスと同様で，ＯＰコード表（前号**別表１**）でインデックスアドレッシングのＯＰコードを調べ，次に**別表２**で目的のポストバイトを調べる。エクステンディドインダイレクトなら，次にアドレスのデータ２バイトをつければいいし，インデックスドインダイレクトなら，必要に応じてオフセットを付け加えればいい。

22 ダイレクトアドレッシング

いろいろなアドレッシングモードが出たが，ここでもう一つ重要なアドレッシングモードを紹介する。ダイレクトアドレッシングと言い，オペランドが１バイトで済む。これが実効アドレスの下位バイトとなり，上位バイトはDP（ダイレクトページレジスタ）の値となる。DPの値の変更は，後述するTFR，EXG，PSH，PULでしかできず，直接値をロードすることはできない。そのため，１回設定するとほとんど変えることはしない。

もともとダイレクトアドレッシングは，6809の前身である6800では上位は00に固定されていた（エクステンドアドレッシングは，ダイレクトよりも広範囲に指定できることからその名が付いた）。それが6809ではDPというレジスタを設け，〝上位バイトはDPの値〟というように改良されたのである。このアドレッシングモードは，速度が速く，命令長も短いので，頻繁に読み出しや書き込みをするワークエリアをこのアドレッシングモードで参照できるアドレスに設定するとよい。なお，FM-８等でもDPを０のままにして，＄０～＄FFをBASICのワークエリアとして使っているはずである。

23 その他のアドレッシングモード

このほかのアドレッシングモードには，インヘレント（オペランドを命令中に含む），アキュムレータがある。どちらもメモリを参照しないので，２つともインヘレントとして扱うことが多い。

24 アドレッシングモードのまとめ

メモリを参照する命令のほとんどは，イミディエイト，ダイレクト，エクステンド，インデックス（インダイレクトを含む）の４つのアドレッシングモードを持つ。この点がほかのCPUと比べて，6809のわかりやすい点である。

また，条件分岐は全てリラティブまたはロングリラティブである。そして種々のメモリからデータを取ってくる必要のない命令がインヘレントになるわけである。

25 無条件分岐・その２

創刊号で紹介したJMP命令には，エクステンドのほか，インデックス，ダイレクトがある。JMP命令が他の命令と違う点は，アドレッシングモードで示されるアドレスが目的のアドレスであるという点で，感覚としては，LEA命令と同一である。つまり実効アドレスがPCに入る。

また，JMP命令でインデックスが使えるということは，BASICでいうON～GOTOに相当する分岐が簡単に行える。

ダイレクトアドレッシングなら，２バイトでジャンプできる。

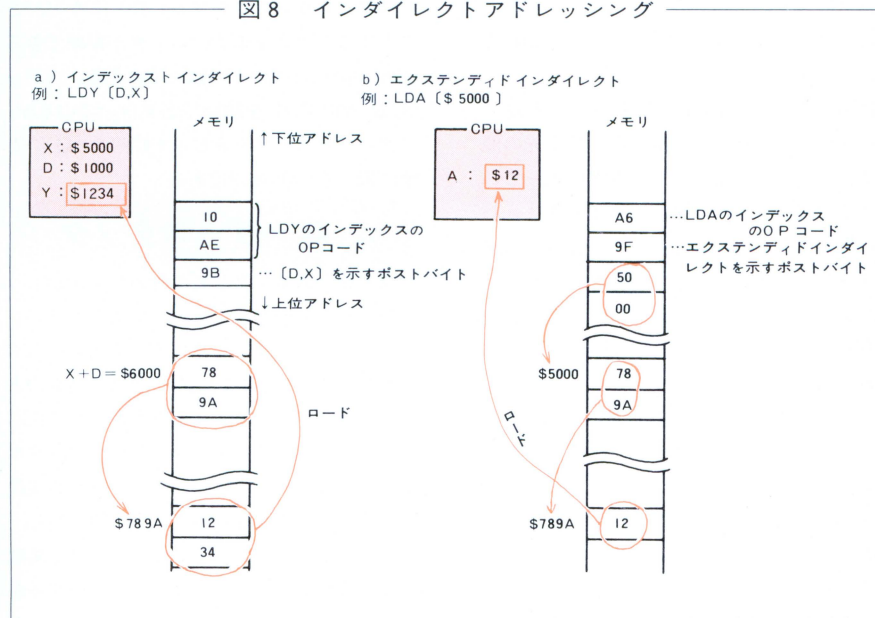
では例を挙げてみよう。

```
JMP ，X
```

なら，Ｘレジスタの値のアドレスに飛ぶ（＝PCの値にＸレジスタの内容を入れる）。

また，

```
JMP 〔＄5000〕
```

図８　インダイレクトアドレッシング

とすれば，意味的には，

LDPC　$5000

（LDPC命令は実際にはないが）という感じである（$5000番地の内容がPCに入る＝$5000番地に書かれているアドレスに飛ぶ）。

26 サブルーチンコール

プログラムの効率を上げるにはサブルーチンを使うのが不可欠であるが，マシン語ではどうなっているのか。これにはJSR（Jamp Sub Routine）命令がある。これはBASICの GOSUBに対応する。動作はJMP同様，実効アドレスに飛ぶが，飛んだ先でリターン命令（RTS: Return from Subroutine）を実行すると，JSRの次の命令に戻ってくる。つまりJSR～RTSはBASICのGOSUB～RETURNに相当する。

今までプログラムの最後にRTSと付けていたのは，BASICのEXECやUSR関数がマシン語をサブルーチンとして呼び出すためである。RTSを実行すればBASICに戻る。

サブルーチンコール命令には，ほかにリラティブモードのBSR，ロングリラティブモードのLBSRがあり，飛び先の指定はBRA，LBRAと同じである。

27 スタック

ところでサブルーチンから戻るというのは，どう実現されているのか。戻り番地がどこかに記憶されていないと不可能である。

実は，これがスタックと呼ばれるメモリエリアに記憶されていく。スタックというのはFILO（First In Last Out）バッファ（データを一時的に蓄えるもの）の一種で，後に入れたデータが先に出てくる。どうしてこうなっていると都合が良いかは，この先を読んでいけば自ずとわかるであろう。

このスタックというバッファは，スタックポインタというレジスタを持つことで実現される。つまり，スタックにデータを入れるときは，スタックをデクリメントしてスタックポインタの示すアドレスにデータを格納する。データを取り出すときはその逆で，スタックポインタの示すアドレスからデータを取り出し，スタックポインタをインクリメントする。一例として，サブルーチンコールを例に挙げると，**図9**のようになる。

このようにスタックはあたかもメモ用紙を積むようにデータを格納することから，この名が付いた。そのため，スタックにデータを格納することを〝スタックに積む〟と言う。

話をサブルーチンコールに戻すと，RTSと

図9　スタック

スタック……積木を積んでいくようにデータを積み上げ，取り出すときは上から取り出す。

いうのはスタックから2バイトデータを取り出してきてPCに入れるわけだ。そのため，スタックポインタをむやみにいじり，RTS実行時にスタックポインタが戻り番地の格納されているアドレスを指していなければ，元のルーチンに戻れなくなる。つまり，関係のないデータでも戻り番地だと見なしてしまうので，スタックポインタを変えるときには，注意が必要である。

スタックはサブルーチンコールにだけ使うわけではない。一時的にレジスタの内容をどこかに退避させたいときにスタックを使えば，一時退避専用のメモリエリアにどこか特定のアドレスを使う必要はない。このレジスタ退避用の命令がPSHSである。PSHSはポストバイトの8つのビットがそれぞれ対応するレジスタを退避させるかどうかを表す。詳しくは前号の**別表4**を見て欲しい。アセンブリ言語で表記するときは，PSHSの後に退避したいレジスタをカンマで区切って書き並べる。

レジスタの格納される優先順位は，PC，U，Y，X，DP，B，A,CCの順で，格納しないからといって，その分スペースは空かず，どんどん詰めて格納されることになる（**図10**参照）。

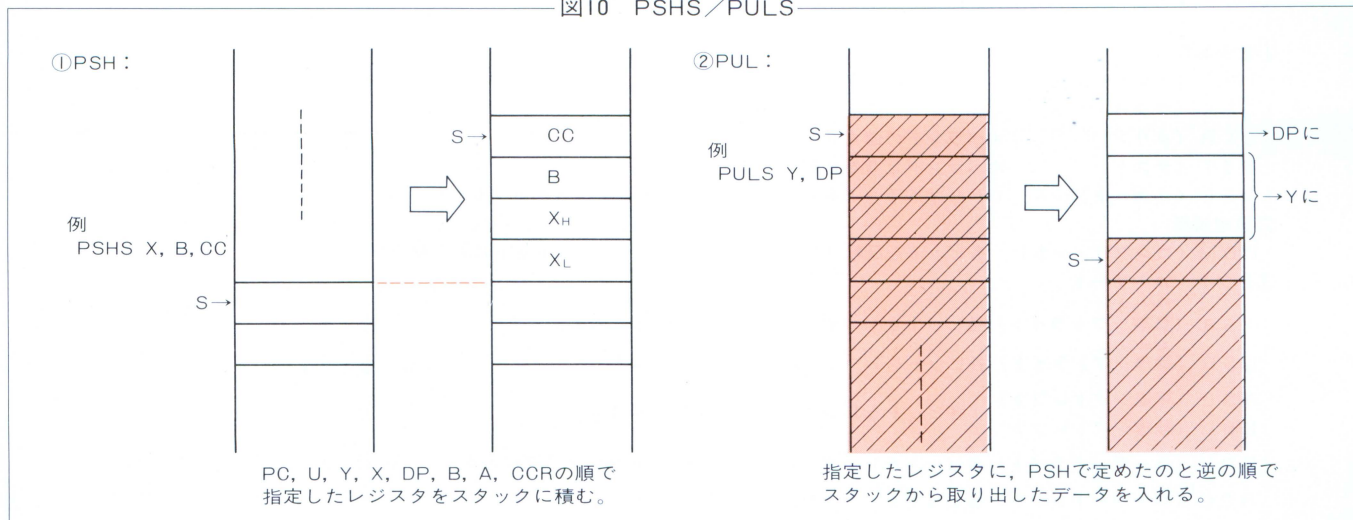
PSHSの反対がPULSである。これはスタックから順にデータを取り出して，先の優先順位と逆順に指定したレジスタに入っていく。この場合も今のスタックポインタにしたがい取り出されるわけで，たとえXレジスタをプッシュ（push：スタックに積むこと）したからといって，プル（pull：スタックから取り出すこと。80系CPUではpopと言う）するときにAレジスタとBレジスタを指定してもよいわけだ。また，PSHS，PULSを用いずとも，1つのレジスタまたは演算結果だけなら，インデックスアドレッシングを使ってデータの読み出し／書き込みはできる。

ポストインクリメント・プリデクリメントもこのスタックの動作に対応している。

前述したように，サブルーチンの開始時と終了時のスタックポインタの値は等しくなければならない。不必要にスタックポインタをインクリメントすると，スタックポインタの示すアドレスより小さいアドレスの内容は，新たにデータがプッシュされると消えてしまう（サブルーチンコールしなくても後述する割り込みがスタックを使う）ので注意してほしい。つまり，スタックを使う際は，常にスタックがいくつ変化したかを頭に入れておく必要がある。

このようにいろいろ制限があるものの，スタックを有効に使えば，ワークエリアがアドレスによらないプログラムが作れるなど，さまざまなプログラムテクニックが駆使できる。6809の優れた点の1つはここにある。スタックは一部のCPUを除いて必ず付いているが，スタックのデータを8ビットCPUで最も柔軟に扱えるのが6809である。

28 ユーザースタック

スタックポインタと名の付いたものがもう1つある。ユーザースタックポインタ（Uレジスタのこと）といい，インデックスレジスタの機能にスタックの機能をプラスしたものと考えてよい。ユーザースタックポインタは，先のスタックポインタ（システムスタックポインタと呼ぶ）がサブルーチンコールや後述の割り込みで値が変わってしまうのに対し，ユーザースタックポインタは，こうした処理の際に値が変わらない。つまり，ユーザーが決めた役割だけ果たし，この点でユーザーが

自由に使える。また，インデックスレジスタ専用にしてもよい。

さて，このスタックポイントに対するプッシュ／プル命令は，それぞれPSHU／PULUである。使い方はPSHS／PULSと同じだが，Uレジスタを退避させる必要がないのでポストバイトのb_6はシステムスタックポインタを指定する役割を持つ。

ユーザースタックポインタの使い方の例として，変数エリアを示すポインタとする手法がある。そしてデータをUレジスタの値から何番目のアドレス，というように指定すれば，Uレジスタの値を変えるだけで変数エリアを変更できる。これによってプログラムの管理がしやすくなる。これとDPとを合わせて考えると，6809の設計思想が伝わってくる。

29　種々の演算命令

創刊号で一部の演算命令を紹介したが，そのほかの演算命令をまとめて説明する。

まず，論理演算（**表5**）から始める。AND，OR，EORの3つはBASICのAND，OR，XORと同じである。これらの命令はアキュムレータとメモリ間の演算で，ADDやSUBなどと同様，A，Bどちらかのアキュムレータを指定する。

（例）　ANDA　＃$20

（アキュムレータAと数値20のANDをとり，結果をアキュムレータAに格納）

このほかの演算命令にBITがある。ANDと同じ演算をするが，結果がアキュムレータに残らず，フラグだけが変わるという命令である。BITとANDの関係は，CMPとSUBの関係と同じである。つまり，あるビットが1か0かを調べるときそのビットだけ1で，他のビットは0の数値とANDをとれば，演算結果により，調べたいビットが1か0かわかる。結果はZフラグに反映されるので，ある特定のビットの状態で条件分けしたいときにBITを使えばよい。

表5　論理演算

AND	…アキュムレータとメモリのANDをとる。結果はアキュムレータに。
OR	……アキュムレータとメモリの OR をとる。結果はアキュムレータに。
EOR	…アキュムレータとメモリのEX-ORをとる。結果はアキュムレータに。
BIT	…アキュムレータとメモリのANDをとる。その結果に応じてフラグをセットする。

このほかに論理演算は特定のビットを無視するときにも使える。

このようにマシン語ではビット操作が単純に行える。

以上がアキュムレータとメモリ間，つまり2つのオペランド*を演算して1つの結果を得る演算である。次に1つのオペランドに対してある操作をする命令に移る。

表6のようにいろいろなものがあり，その演算対象，すなわちオペランドは，アキュムレータA，B，メモリが可能で，メモリの場合は，4つのアドレッシングモードのうちイミディエイト以外すべてが可能である。AまたはBレジスタの場合は，

INCA
ROLA

などとする。

さて，**表6**-①の命令のうち，INC，DECは，ADDやSUBとフラグの変化が異なる点に注意してほしい（創刊号の**別表1**参照)。

TST命令は，オペランドの内容でNフラグとZフラグを変化させる。つまりオペランドのb_7がNフラグに入る。他にもb_7は他のビットと比べて調べやすいのでb_7をうまく使うのもテクニックの1つである。

NEGは符号付きの数の符号反転(－1倍)に便利である。また，Dレジスタの NEGはないが，

NEGA
NEGB
SBCA　＃$00

で実現できる。

シフト命令には算術シフト（arithmetic shift）と論理シフト（logical shift）がある。なぜこういう分類があるかというと，算術シフトは符号を考えてシフトする。シフトというのは2進数で一桁ずらすのだから，オペランドを数値として扱った場合，左シフトは2倍，右シフトは½を意味することになる。このとき，オペランドが符号付きの整数なら，符号のことも考えて欲しいわけである。そのため，ただ各ビットをずらすだけの論理シフトとは，右シフトの算術シフトは異なる。つまり**表6**の動作説明のようにすれば，符号も考慮されたことになる。なお左シフトのときは，符号が保存されない（つまりオーバーフローを起こす）場合，フラグに反映されるようになっている。

＊6809の命令表では，レジスタをオペランドとせず命令中に含めるが，ここではレジスタもオペランドとして扱っている。したがって表6の各命令の動作説明もアキュムレータA，Bを含めてオペランドをMで代表させた。

表6 いろいろな1オペランド命令

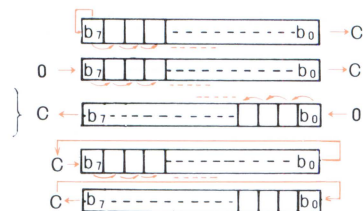
①数値演算

命令		内容	
INC	(インクリメント)	オペランドの値を1つ増やす(＋1)	M＋1→M
DEC	(デクリメント)	オペランドの値を1つ減らす(－1)	M－1→M
CLR	(クリア)	オペランドの値を0にする	0→M
TST	(テスト)	オペランドと0を比較	M：0
NEG	(ネガート)	0からオペランドを引き，結果をオペランドに入れる	0－M→M

②論理演算

命令		内容	
COM	(コンプリメント)	各ビットを反転(1のビットを0に，0のビットを1に)	$\overline{M}$→M

③シフト・ローテート

命令		内容
ASR	(算術(アリスメティック)シフトライト)	各ビットを1つずつ右にずらす。b_7はそのまま
LSR	(論理(ロジカル)シフトライト)	各ビットを1つずつ右にずらす。b_7は0
ASL	(算術シフトレフト)	各ビットを1つずつ左にずらす。b_0は0
LSL	(論理シフトレフト)	各ビットを1つずつ左にずらす。b_0は0
ROR	(ローテートライト)	各ビットを1つずつ右に。b_7にはキャリーフラグが入る
ROL	(ローテートレフト)	各ビットを1つずつ左に。b_0にはキャリーフラグが入る

数値のシフトでも，符号なしなら，各ビットをずらすだけの論理シフトを使う。なお，LSLはASLで兼ねられるので，同一のOPコードとなっている。

ローテート命令は他のシフト命令と合わせて，多倍長のシフトをするときに用いる。例えば，Dレジスタのシフトはないが，

```
ASLB
ROLA
```

とすれば，DレジスタのASLが実現できる。また，1命令でキャリーフラグの内容をメモリに入れたいときにも使える。

さて，これらのシフト命令は，このようにシフトさせることによって，乗算や除算に使う。ただ6809の場合，乗算命令があるので，シフトを使わずに乗算ができる。乗・除算のアルゴリズムは，ここでは詳しく触れられないが，符号付きの場合は，絶対値で演算し，結果の符号はあとで考えなければならない。

30 種々の特殊な命令

続いて，種々の特殊な命令に話を進める。**表7**を見て欲しい。

表7　種々の特殊命令

命令	内容	
ABX	XレジスタにBレジスタの内容を符号なしとして加算	X＋B→X
BRN	条件分岐命令だが，分岐の条件は常に不成立(BRanch Never)	
MUL	AレジスタとBレジスタを乗算し，結果をDレジスタに入れる(符号なし)	A×B→D
NOP	何もしない	No Operation
SEX	Bレジスタの値をDレジスタに入れる(符号つき)	
EXG R_1, R_2	R_1の内容とR_2の内容を交換	R_1←→R_2
TFR R_1, R_2	R_1の内容をR_2に入れる	R_1──→R_2
ANDCC	イミディエイト値とコンディションコードレジスタをAND	CC∧M→CC
ORCC	イミディエイト値とコンディションコードレジスタをOR	CC∨M→CC

まずABXであるが，これは表のとおりだ。LEAX　B，Xとは符号なしという点と，実行速度が速いという点が異なる。MULも表のとおりである。NOPは，何の処理もしない命令である。用途としては，プログラムを部分的に変更したいとき変更の前後でプログラムのバイト数が変わってしまっては不都合が多い。こういう場合にNOPで調整できる。BRNもこれと同様で，条件分岐を一時的に中止するときなどに有効である。しかし使うことはほとんどない。

SEX(Sign EXtend)はBレジスタに入っている数値を符号付きと考え，8ビットの符号付きを16ビットの符号付きに変換した結果をDレジスタに入れる。つまりBレジスタのb_7の値でAレジスタの全ビットを埋めるわけだ。

EXGとTFRは，8ビットのレジスタ間，または16ビットのレジスタ間のデータ転送命令である。EXGは2つのレジスタの値を入れかえる(BASICのSWAPと思えばいい)。TFRは一方のレジスタの値をもう一方に入れるだけである。さて，その2つのレジスタの指定はポストバイトによって行う。各レジスタを4ビットのコードで表現し，ポストバイトの上位4ビットが転送元のレジスタを，下位4ビットが転送先のレジスタを示す。詳しくは前号**別表3**(2進表記)か，今号**別表7**(16進表記)を見てほしい。

ANDCC,ORCCはコンディションコードレジスタを変更する命令で，これはイミディエイトのデータとCCRの内容で，それぞれAND，ORをし，結果をCCRに入れる。つまり，CCRの特定のビット，すなわち特定のフラグを0にしたり1にしたりできる。1にしたいときは目的のビットのみ1で，他のビットは0の値とORを取ればよく，0にしたいときは目的のビットのみ0で，他のビットは1の値とANDを取ればよいというわけだ。

〔別表7〕EXG/TFRポストバイト

レジスタ	コード	レジスタ	コード
D	0	PC	5
X	1	A	8
Y	2	B	9
U	3	CC	A
S	4	DP	B

TFR…上位→下位　EXG…上位↔下位

31 割り込み

6809マシン語入門のラストは割り込み(Interrupt：インタラプト)である。コンピュータの処理するものには，いつ処理が必要になるかわからないものもある。たとえていえば，かかってくる電話のようなものだ。つまり，いつかかってくるかわからない電話を待っていては他の仕事ができないし，かといって仕事の方ばかり熱中しては電話に出られない。人間は電話のベルが鳴るということで電話の応対が必要であるということを知ること

図11 割り込み

NMI ——→すべてのレジスタをプッシュ：E＝1，I＝1，F＝1にする→$FFFCに入っているスタートアドレスに飛ぶ

IRQ ——→Iフラグは0？ —YES→ すべてのレジスタをプッシュ：E＝1，I＝1 にする——→$FFF8に入っているスタートアドレスに飛ぶ

FIRQ ——→Fフラグは0？ —YES→ PCとCCのみプッシュ：E＝1，I＝1，F＝1にする——→$FFF6に入っているスタートアドレスに飛ぶ

RESET ——→$FFFEに入っているスタートアドレスに飛ぶ

SWI 実行→すべてのレジスタプッシュ：E＝1，I＝1，F＝1にする——→$FFFAに入っているスタートアドレスに飛ぶ

SWI2実行→すべてのレジスタプッシュ：E＝1にする——→$FFF4に入っているスタートアドレスに飛ぶ

SWI3実行→すべてのレジスタプッシュ：E＝1にする——→$FFF2に入っているスタートアドレスに飛ぶ

ができ，いったん仕事を中断して電話の応対が可能である。マイコンも同様で，外部から特別の処理が必要になるとき，特定の信号をCPUに送ることによって現在行っている処理をいったん中断して，別の処理をさせることができる。

これが割り込み処理で，緊急に必要になった処理を済ませたら，またもとの処理を続ける。

さて，割り込み処理はどのように行われるのであろうか。**図11**を見てほしい。

割り込みにはNMI（ノンマスカブルインタラプト），IRQ（インタラプトリクエスタブル），FIRQ（Fast IRQ）の3種がある。

IRQ，FIRQはそれぞれCCRのIフラグ，FフラグをORCC命令などで1にすることで，割り込みの信号がきても割り込み処理は始まらない。

外部から割り込み要求の信号が送られると割り込み処理をしてよいかCCRのフラグを調べる。割り込み可となると，レジスタの値をスタックに積む（NMI，IRQはすべてのレジスタ，FIRQはPCとCCRのみ）。そしてアドレス空間の終わりの方にある割り込みベクトル（割り込み処理開始アドレス）を読み込んで，そのアドレスにジャンプする。そこに必要な処理をするプログラムをおけばよい。

処理が終わればRTI（Return from Inter-rupt）を実行して，もとの処理を再開させる。これは，

```
PULS  PC, U, Y, X, DP, B, A, CC
```

と同じ処理になる。ただし，FIRQの場合は，PCとCCのみなので，

```
PULS  PC, CC
```

とならなくてはいけない。これはCCRのEフラグによって判断される。すなわち，NMI，IRQで割り込み処理が始まったときはEを1に，FIRQならEを0にする。RTIはEフラグを見て，どのレジスタをプルするか決める。

また，割り込みに似たものにリセットがある。これはRESET信号をCPUに送ると動作し，今までのことをすべてご破算にして，所定のアドレスから命令を実行する。電源投入時や，暴走時などに使う。

こうした割り込みはI／Oの管理やタイマ割り込みなどに使われる。

以上が外部からの割り込みであるが，割り込みと同じ動作をスタートさせる命令がある。奇妙ではあるが，サブルーチンコールと割り込みのあいの子のようなもので，活用の仕方次第では非常に有効なものである。動作は**図11**を見てもらえばわかると思う。

用途としてはSWI処理ルーチンでモニタなどにジャンプさせれば，スタックに格納された各レジスタの値を調べられる。こうすればデバッグ中にあるアドレスでプログラムをいったん停止させたい場合，そのアドレスにこの命令を1バイト入れるだけでよい。

さて，こうした割り込みには優先順位があり，それに応じてIフラグ，Fフラグをセットし，下位の割り込みを禁止するようになっている。

割り込みに関する命令として，SYNC，CWAIがある。SYNCは割り込みに対してすばやく処理するためのもので，これを実行すると割り込み信号が送られてくるまで待つ。割り込み信号が送られればあとは通常の動作に移る。割り込み不可なら次の命令を，可なら割り込み処理を実行する。つまり，割り込みをマスクして使えば，外部の信号に同期した処理が条件分岐なしでできる。

CWAIは，コンディションコードレジスタとイミディエイト値のANDをとって割り込みを待つ。割り込みのマスクの解除と割り込み待ちがまとめて可能となる。

32 おわりに

以上で6809のマシン語命令の解説は，終わりである。他のCPUと比べ豊富なアドレッシングモードと対称性がよく使いやすい命令群，豊富な条件分岐，強力なスタックと，8ビットCPUとしては最高級で，命令も統一的に扱っているため，59種と覚えるのが楽である。さらに相対分岐，プログラムカウンタリラティブを使ってアドレスとは無関係なプログラム（位置独立プログラム）が作れる。さらに，スタックをワークエリアとして使えることによってリエントラントなプログラム，リカーシブなプログラムが簡単につくれる。

リエントラントなプログラムとは，実行を中断して，またそのプログラムを実行しても，データが壊れないために再開できるプログラムのことである。たとえば割り込み処理中で呼び出すサブルーチンなどは，通常の処理で使うものでもリエントラントでないと都合が悪い。

リカーシブなプログラムとは，自分自身をサブルーチンコールできるようなプログラムのことである。

今回は一とおりの命令の解説しかできなかったが，機会があればいろいろなテクニックなどについて詳しく触れたいと思う。

参考文献

・インターフェース，'81，2月号
・アスキー，'80，10月号
・日立8ビットマイクロコンピュータデータマニュアル（'82）

for FM-7 FM-8

高速3Dグラフィックス

田中　明雄

オールアセンブラで，高速にメッシュ図形を描くプログラムを紹介する。スピードは，BASICの数十倍である。

アセンブラがグラフィックに適しているわけでは決してない。ただ，パソコン程度のコンピュータでスピードを求めるには，いきおいコンパイラなりアセンブラなりを使わざるを得ないだろう（余談だがYHPのHP9000では，BASICでワイヤフレームで描いた家を，リアルタイムで回転させられる）。

今回はアセンブラを勉強しはじめたばかりの人を対象に，ソースリストを使って説明する。

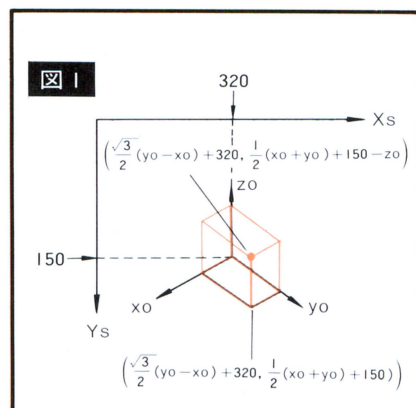
座標

汎用パッケージではないので，座標軸を図1のように固定し，等軸，平行投影とする。すると対象座標系（表示する図形の座標系）からスクリーン座標系（画面の横640，縦200の座標）への変換は，次のように簡単に表せる。

$$\begin{cases} Xs=\frac{\sqrt{3}}{2}(Y0-X0)+320 \\ Ys=\frac{1}{2}(X0+Y0)-Z0+150 \end{cases}$$

ボックスを描くとわかるがFM-7，8も含めて，たいていのパソコンは縦と横の比（アスペクト比）が1：1ではない。LINE(0，0)－(100，100)，PSET，7，Bでは，正方形ではなく長方形になってしまう。大型のディスプレイシステムの中には，これを変更できるものもある。残念ながらFM-7，8にはこの機能はないので，プログラム中で補正する。結局，先の変換式を次のような簡単な形にして使う。

$$\begin{cases} Xs=(Y0-X0)+320 \\ Ys=(X0+Y0)/4-Z0+150 \end{cases}$$

関数

今回は次の関数を使った。

Z＝（1＋COS(X) COS(Y)）

この関数の計算部分を変更すれば，他の関数を表示することも可能である（ただし汎用パッケージではないので，容易に，とはいえない）。Z＝（1＋COS(X) COS(Y)）は，よくデモなどで使われている，キャップのような形状をしている（7ページの写真参照）。本プログラムでは変化をつけるために，Zの高さによって色を変えるようにした。

COSの値はシングルテーブルを引くことで高速化した（テーラー展開の必要はないだろう）。テーブルは0°～90°のCOS値の128倍のデータにした。128＝2^7倍にしたのは，不要な除算を避けるためである。2のベキ乗の除算なら，シフトで行える。こういったことはフライトシミュレータなどスピードの要求されるプログラムでは，多用されている。

メッシュ

メッシュとは網のことである。曲面を表現する方法としては，もっとも単純な方法である。最近のCG(コンピュータグラフィック)では，ソリッドモデルが中心で，ワイヤフレームやメッシュによる出力は，それを援助する目的で使われている感が強い。

メッシュの方法は，あるYに対してXを変化させ，Zをプロットしていく。これをX，Y逆にして行えば，網(メッシュ)となる。

PSET

点は，FBIOSをコールしてサブシステムにやってもらう。サブシステムへは一度に最大20ドットの情報を渡せる。そのためプログラムでは，点が20ドットたまった時点で，F-BIOSをコールするようになっている。こうすることにより，FBIOSコールによるオーバヘッドが極力おさえられる。FBIOSコールによりサブルーチンを呼びだす場合，ユーザーの作ったデータを共有RAM領域へ転送して処理されるわけで，たびたびやっていたのでは実行速度が落ちてしまう。FBIOS，サブシステムのコールの方法は前回にも説明があったが，プログラム中でPSETについて解説しよう。

プログラム

では，プログラムの説明に入っていく。重要な部分をかいつまんで説明しよう。

〈CUZCRD〉

これは（X，Y）からZを計算する。

LDA XCRD ： AレジスタにX座標をロード

LBSR COS ： COSルーチンをコール

EXG A，B ： COS(X)をBレジスタに退避

LDA YCRD ： AレジスタにY座標をロード

LBSR COS ： COSルーチンをコール

MUL：COS(X)＊COS(Y)

以上でDレジスタに，COS(X)＊COS(Y)の128＊128倍がセットされる。これを128で割るには

ASLA：Aレジスタを左へ算術シフト

EXG A,B：AとBを入れ換える

SEX：Bレジスタを符号拡張

すればよい。先述したように2のベキ乗による除算（乗算も）はシフトで行える。

ADDD ＃128：Dレジスタに128を加える

Dレジスタをシフトする命令はないので，AレジスタとBレジスタのシフトとローテートで行う（マクロ命令の使えるアセンブラなら，LSRDなどを定義しておくと便利）。

LSRA ： Aレジスタを右へ論理シフト（最下位ビットはキャリーへ）

RORB ： Bレジスタを右にローテート（キャリーはBのMSBに入る）

これでDレジスタに関数の値がセットできたので，

STD ZCRD

RTS

で終了となる。

〈CULCOL〉

Z座標によってカラーコードをセットするルーチンである。

Z＜＝$49 →Color＝2

$49＜Z＜＝$60 →Color＝3

Z＞$60 →Color＝7

となる。

LDD ZCRD ： DレジスタにZ座標をロード

CMPD ＃$0060 ┐ もしZ＜＝$60ならば
BLE CULC 1 ┘ CULC 1へ

LDA ＃7 ┐
STA COLOR │ COLOR＝7にしてリターン
RTS ┘

・CULC 1

CMPD ＃$49 ┐ もしZ＜＝$49ならば
BLE CULC 2 ┘ CULC 2へ

LDA ＃3 ┐
STA COLOR │ COLOR＝3にしてリターン
RTS ┘

・CULC2

LDA ＃2 ┐
STA COLOR │ COLOR＝2にしてリターン
RTS ┘

〈CUXSCR〉

X, Y座標からスクリーン座標に直すもの。

XSCR＝(Y－X)＋320

を計算する。

LDB XCRD ┐ XCRDをBレジスタに
SEX ┘ ロードして符号拡張

XCRDはバイト長であるのに対し，XSCRはワードデータなので，符号拡張が必要。

STD BUFFER：X座標をBUFFERに退避

LDB YCRD ┐ YCRDをロードし
SEX ┘ 符号拡張

SUBD BUFFER：Y－X

ADDD ＃320：(Y－X)＋320

STD XSCR ┐ DレジスタをXSCRに
RTS ┘ セットして終了

〈CUYSCR〉

CUXSCRと同様，スクリーン座標系のY座標を計算する。

LDB XCRD ┐
SEX ┘ XCRDを符号拡張

STD BUFFER：BUFFERに退避

LDB YCRD ┐
SEX ┘ YCRDを符号拡張

ADDD BUFFER：Dレジスタ←(X＋Y)

Dレジスタを4で割るには先述のシフトを2回行う。

ASRA

RORB

ASRA

RORB

SUBD ZCRD：Dレジスタ＝(X＋Y)／4－Z

ADDD ＃150：Dレジスタ＝(X＋Y)／4－Z＋150

STD YSCR ┐
RTS ┘ YSCRにセットして終了

〈PLOT〉

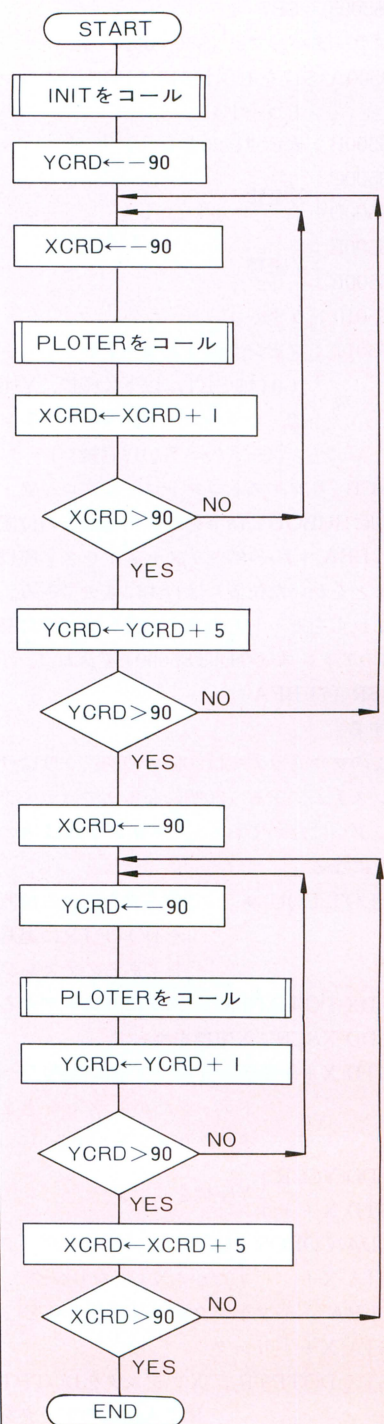
FBIOS経由で点を打つには，次のようにする。

・$5000に16をセット（16はSUBOUT リクエスト番号）

・$5002，$5003にデータバッファアドレス$5008をセット（$5002）＝$50，（$5003）＝$08

・$5004，$5005にデータバイト数をセット。

MAINルーチンフローチャート

START
INITをコール
YCRD←－90
XCRD←－90
PLOTERをコール
XCRD←XCRD＋1
XCRD＞90 （NO → XCRD←－90の後へ）
YES
YCRD←YCRD＋5
YCRD＞90 （NO → XCRD←－90の前へ）
YES
XCRD←－90
YCRD←－90
PLOTERをコール
YCRD←YCRD＋1
YCRD＞90 （NO → YCRD←－90の後へ）
YES
XCRD←XCRD＋5
XCRD＞90 （NO → YCRD←－90の前へ）
YES
END

今回は123バイト転送だから($5004)＝0，($5005)＝$B7

次にデータバッファ（$5008以降）は

$500A：$17をセット($17はPOINTのコマンドコード）

$500B：表示点数20をセット

$500C, $500D：X座標

$500E, $500F：Y座標

$5010：カラーコード

$5011：ファンクションコード0をセット（0はPSET，以下X座標，Y座標，カラーコード，ファンクションコードのペアが19個続く）

RCB（リクエストコントロールブロック；最初にSUBOUT命令をかいた場所）の位置や，DBA（データバッファアドレス；座標値などをかいた位置）は自由に変更できる。

これらをセットしたあと，XレジスタにRCBのアドレス（今回は$5000）を入れて，

JSR〔$FBFA〕

とする。

このプログラムでは20個の点を，一度にサブシステムに送る（20個になるまではデータバッファにためておく）。プログラムは次のようになる。

```
PLOT EQU ＊
LDX DOTPTTR      データを入れる位置をDOTPTRに退避してあるのでXレジスタにリストアする
LDD XSCR：X座標をロード
STD X＋＋：データバッファに格納(ワードデータだから X＋＋ と2インクリメント）する
LDD YSCR
STD X＋＋        Y座標をリンク
LDA COLOR
STA X＋          COLOR（バイトデータだからX＋）をリンク
CLRA
STA X＋          ファンクションコード0をリンク
STX DOTPTR： XレジスタをDOTPTRに入れ次のデータを格納する位置を保存する
DEC PCOUNT： PCOUNT(バッファに登録された点の個数）をデクリメントして
BEQ CALBIOS： 20個になったら(PCOUNT＝0となったら）CALBIOSへ
RTS：さもなければリターン
CALBIOS EQU ＊
LDA ＃20
STA PCOUNT       デクリメントして0になったものを20にリセットする
LDD ＃DBA＋4
STD DOTPTR： データの格納場所を最初の位置へ
LDX ＃RCB： XレジスタにRCBの位置を入れる
JSR〔FBIOS〕：FBIOS をコール
RTS：終了
```

〈COS〉

COSをテーブル参照で求めるプログラムである。COSのテーブルはCOSTBL以降にあるものとする。入力データは，−90°〜＋90°とし，出力はCOSの128倍を返す（バイトデータ）。

```
LDX ＃COSTBOL：Xレジスタにテーブルのトップをロード
TSTA
BGE COS1         Aレジスタの絶対値をとる
NEGA             (∴COS(−X)＝COS(X))
●COS1 LDA A, X： Xレジスタ間接でAにロードすると簡単にCOS値がセットされる
RTS：終了
```

〈INIT〉

RCBとデータバッファを初期化するルーチンで，PLOTで説明したようにデータをセットする。

```
INIT EQU＊
LDX ＃RCB：XレジスタにRCBのアドレスをロード
LDA ＃16
STA X＋          PSETのコマンドをセット
CLRA
STA X＋
LDD ＃DBA：Dレジスタにデータバッファアドレスをロード
STD X＋＋
LDD ＃123
STD X＋＋        123＝転送データ数をセット
LDX＃DBA＋2：Xにデータバッファアドレス＋2をセット
LDA ＃$17        pointのコマンドコード
STA X＋          $17をセット
LDA ＃20
STA X＋          20（転送データ数）をセット
STX DOTPTR：データ格納位置を保存
RTS：終了
```

〈PLOTER〉

X座標，Y座標をセットすると自動的に点を打つルーチンである。

```
BSR CUZCRD：Zを計算
BSR CULCOL：色をセット
BSR CUXSCR：X座標をセット
BSR CUYSCR：Y座標をセット
BSR PLOT：プリミティブな点を打つルーチンをコール
RTS：終了
```

〈MAIN〉

以上のルーチンを使って，メインルーチンを書く。イニシャライズ，X軸方向のメッシュ，Y軸方向のメッシュを描くルーチンにわかれている。

```
MAIN EQU ＊
     BSR INIT：イニシャライズ
・X軸方向のメッシュ
YDIREC LDA ＃−90
       STA YCRD       Y座標を−90に
YDIRL1 LDA ＃−90
       STA XCRD       X座標を−90に
YDIRL2 BSR PLOTER：点を打つ
       LDA XCRD
       ADDA ＃1
       STA XCRD       X座標を1増す
       CMPA ＃90
       BLE YDIRL2     X座標が90以下ならばYDIRL2へ
       LDA YCRD
       ADDA ＃5
       STA YCRD       Y座標を5増す
       CMPA ＃90
       BLE YDIRL1     Y座標が90以下ならばYDIRL1へ
```

以上のルーチンをX座標，Y座標入れ換えると，次のXDIRECになる。

```
XDIREC EQU＊
     ∫同様のルーチンなので説明略
RTS：終了
```

〈最後に〉

プログラムの開発はFM-8のFLEX®で行った。アセンブラによる開発には，やはりOSが完備していることが望ましい。

実行速度はアセンブラだけあって，まあまあなものになった。BASICプログラムの場合と比較してほしい。最後にCOSTBLを作るプログラムを載せておく。

FLEX®は，TSC社の登録商標です。

```
                    ****************************************
                    *                                      *
                    *  3D HIGH-SPEED GRAPHICS              *
                    *                                      *
                    ****************************************
```



```
            FBFA    FBIOS   EQU     $FBFA
            5000    WORK    EQU     $5000

4000                        ORG     $4000
            4000    MAIN    EQU     *
                    *
                    * MAIN ROUTINE FOR 3D GRAPHICS
                    *
4000 8D     49              BSR     INIT        CALL INITIALIZE ROUTINE

4002 86     A6      YDIREC  LDA     #-90
4004 B7     5091            STA     YCRD
4007 86     A6      YDIRL1  LDA     #-90
4009 B7     5090            STA     XCRD
400C 8D     60      YDIRL2  BSR     PLOTER

400E B6     5090            LDA     XCRD        INCREMENT X BY 1
4011 8B     01              ADDA    #1
4013 B7     5090            STA     XCRD

4016 81     5A              CMPA    #90         X>90 ?
4018 2F     F2              BLE     YDIRL2

401A B6     5091            LDA     YCRD        INCREMENT Y BY 5
401D 8B     05              ADDA    #5
401F B7     5091            STA     YCRD

4022 81     5A              CMPA    #90         Y>90 ?
4024 2F     E1              BLE     YDIRL1      IF Y<=90 THEN YDIRL1

4026 86     A6      XDIREC  LDA     #-90
4028 B7     5090            STA     XCRD        X=-90
402B 86     A6      XDIRL1  LDA     #-90
402D B7     5091            STA     YCRD        Y=-90
4030 8D     3C      XDIRL2  BSR     PLOTER

4032 B6     5091            LDA     YCRD        INCREMENT Y BY 1
4035 8B     01              ADDA    #1
4037 B7     5091            STA     YCRD
403A 81     5A              CMPA    #90         Y>90 ?
403C 2F     F2              BLE     XDIRL2      IF Y<=90 THEN XDIRL2

403E B6     5090            LDA     XCRD        INCREMENT X BY 5
4041 8B     05              ADDA    #5
4043 B7     5090            STA     XCRD

4046 81     5A              CMPA    #90         X>90 ?
4048 2F     E1              BLE     XDIRL1      IF X<=90 THEN XDIRL1

404A 39                     RTS

                    *
                    * END OF MAIN ROUTINE
                    *

            404B    INIT    EQU     *
                    *
                    * INITIALIZE ROUTINE
                    *
404B 8E     5000            LDX     #RCB        X=RCB TOP ADRESS
404E 86     10              LDA     #16         16 REQUEST NUMBER OF POINT (PSET)
4050 A7     80              STA     X+

4052 4F                     CLRA
4053 A7     80              STA     X+
```

```
4055 CC    5008              LDD    #DBA       D=DBA ( DATA BUFFER TOP ADRESS )
4058 ED    81                STD    X++

405A CC    007B              LDD    #123       123=TRANSFER DATA NUMBER
405D ED    81                STD    X++

405F 8E    500A              LDX    #DBA+2     X=DATA BUFFER TOP ADRESS+2

4062 86    17                LDA    #$17       $17 =COMMAND CODE OF 'POINT'
4064 A7    80                STA    X+

4066 86    14                LDA    #20        20 TRANSFER POINTS NUMBER
4068 A7    80                STA    X+

406A BF    5088              STX    DOTPTR     BUFFERING DOT-POINTER

406D 39                      RTS

               406E  PLOTER  EQU    *
                     *
                     * PAN-PLOTTING ROUTINE
                     *
406E 8D    09                BSR    CUZCRD     CUL-Z-CO-ORDINATE
4070 8D    23                BSR    CULCOL     CUL-COLOR
4072 8D    42                BSR    CUXSCR     CUL-X-SCREEN CO-ORDINATE
4074 8D    55                BSR    CUYSCR     CUL-Y-SCREEN CO-ORDINATE
4076 8D    6F                BSR    PLOT       PRIMITIVE PLOT ROUTINE

4078 39                      RTS

                     * END OF PLOTER SUBROUTINE
                     *
               4079  CUZCRD  EQU    *
                     *
                     * CUL-Z-CO-ORDINATE
                     *
                     * THIS ROUTINE DETERMINE 'IMAGE'
                     *
                     * Z=(1+COS(X)*COS(Y))
                     *
4079 B6    5090              LDA    XCRD
407C 17    0099              LBSR   COS        A.REG=COS(X)
407F 1E    89                EXG    A,B        BUFFERING COS(X) IN B.REG

4081 B6    5091              LDA    YCRD
4084 17    0091              LBSR   COS        A.REG=COS(Y)

4087 3D                      MUL               D.REG=COS(X)*COS(Y)
4088 48                      ASLA              A.REG=D.REG/128
4089 1E    89                EXG    A,B        EXCHANGE A,B FOR SIGN EXTEND
408B 1D                      SEX               THE MOST SENSITIVE OP-CODE OF 6809

408C C3    0080              ADDD   #128       D.REG=128+COS(X)*COS(Y)/128

408F 44                      LSRA              D.REG=D.REG/2
4090 56                      RORB
4091 FD    5092              STD    ZCRD       SET Z-CO-ORDINATE

4094 39                      RTS

               4095  CULCOL  EQU    *
                     *
                     * CUL-COLOR
                     *
4095 FC    5092              LDD    ZCRD
4098 1083  0060              CMPD   #$0060     Z>$0060 ?
409C 2F    06                BLE    CULC1

409E 86    07                LDA    #7         IF Z>$60 THEN COLOR=7
```

```
40A0 B7    508E               STA    COLOR

40A3 39                       RTS

40A4 1083 0049      CULC1     CMPD   #$49       Z>$49 ?
40A8 2F    06                 BLE    CULC2

40AA 86    03                 LDA    #3
40AC B7    508E               STA    COLOR      IF $49<Z<=$60 THEN COLOR=3
40AF 39                       RTS

40B0 86    02       CULC2     LDA    #2
40B2 B7    508E               STA    COLOR      IF Z<=$49 THEN COLOR=2
40B5 39                       RTS

              40B6  CUXSCR    EQU    *
                    *
                    * CUL-X SCREEN CO-ORDINATE
                    *
40B6 F6    5090               LDB    XCRD       SIGN EXTEND XCRD
40B9 1D                       SEX
40BA FD    5094               STD    BUFFER     BUFFERIN' X

40BD F6    5091               LDB    YCRD       SIGN EXTEND YCRD
40C0 1D                       SEX

40C1 B3    5094               SUBD   BUFFER     D.REG=Y-X ( WORD )
40C4 C3    0140               ADDD   #320       D.REG=(Y-X)+320
40C7 FD    508A               STD    XSCR       SET X-SCREEN CO-ORDINATE

40CA 39                       RTS

              40CB  CUYSCR    EQU    *
                    *
                    * CUL-Y-SCREEN CO-ORDINATE
                    *
40CB F6    5090               LDB    XCRD
40CE 1D                       SEX
40CF FD    5094               STD    BUFFER
40D2 F6    5091               LDB    YCRD
40D5 1D                       SEX
40D6 F3    5094               ADDD   BUFFER     D.REG=X+Y ( WORD )

40D9 47                       ASRA              D.REG=D.REG/4
40DA 56                       RORB
40DB 47                       ASRA
40DC 56                       RORB
40DD B3    5092               SUBD   ZCRD       D.REG=(X+Y)/4-Z
40E0 C3    0096               ADDD   #150
40E3 FD    508C               STD    YSCR       Y=(X+Y)/4-Z+150

40E6 39                       RTS

              40E7  PLOT      EQU    *
40E7 BE    5088               LDX    DOTPTR     X=DOT POINTER
40EA FC    508A               LDD    XSCR
40ED ED    81                 STD    X++        LINK X-SCREEN ADRESS

40EF FC    508C               LDD    YSCR
40F2 ED    81                 STD    X++        LINK Y-SCREEN ADRESS

40F4 B6    508E               LDA    COLOR
40F7 A7    80                 STA    X+         LINK COLOR CODE

40F9 4F                       CLRA
40FA A7    80                 STA    X+         LINK FUNCTION CODE ( 0=PSET )

40FC BF    5088               STX    DOTPTR     BUFFERING DOT-POINTER
40FF 7A    508F               DEC    PCOUNT     DECRIMENT POINT COUNTER
4102 27    01                 BEQ    CALBIOS    IF PCOUNT=0 THEN CALL BIOS

4104 39                       RTS
```

```
               4105  CALBIOS EQU     *
 4105 86    14               LDA     #20
 4107 B7    508F             STA     PCOUNT     SET POINT COUNTER=20

 410A CC    500C             LDD     #DBA+4
 410D FD    5088             STD     DOTPTR     RESTORE DOT-POINTER

 4110 8E    5000             LDX     #RCB
 4113 AD    9F FBFA          JSR     [FBIOS]    CALL FBIOS
 4117 39                     RTS

               4118  COS     EQU     *
                     *
                     * COS SUBROUTINE
                     * INPUT A-REG ( DEGREE ) -90<A.REG<90
                     * OUTPUT A.REG ( 128*COS(A.REG) )
                     *
 4118 8E    5096             LDX     #COSTBL
 411B 4D                     TSTA
 411C 2C    01               BGE     COS1       A.REG=ABS(A.REG)

 411E 40                     NEGA

 411F A6    86       COS1    LDA     A,X

 4121 39                     RTS

 5000                        ORG     $5000

 5000                RCB     RMB     8          RCB REQUIRE 8 BYTES AREA
 5008                DBA     RMB     128        DATA BUFFER AREA

 5088 0000           DOTPTR  FDB     0
 508A 0000           XSCR    FDB     0
 508C 0000           YSCR    FDB     0
 508E 07             COLOR   FCB     7
 508F 14             PCOUNT  FCB     20

 5090 00             XCRD    FCB     0
 5091 00             YCRD    FCB     0
 5092 0000           ZCRD    FDB     0

 5094 0000           BUFFER  FDB     0
 5096                COSTBL  RMB     91

                             END

0 ERROR(S) DETECTED

SYMBOL TABLE:

BUFFER 5094   CALBIO 4105   COLOR  508E   COS    4118   COS1   411F
COSTBL 5096   CULC1  40A4   CULC2  40B0   CULCOL 4095   CUXSCR 40B6
CUYSCR 40CB   CUZCRD 4079   DBA    5008   DOTPTR 5088   FBIOS  FBFA
INIT   404B   MAIN   4000   PCOUNT 508F   PLOT   40E7   PLOTER 406E
RCB    5000   WORK   5000   XCRD   5090   XDIREC 4026   XDIRL1 402B
XDIRL2 4030   XSCR   508A   YCRD   5091   YDIREC 4002   YDIRL1 4007
YDIRL2 400C   YSCR   508C   ZCRD   5092
```

COSテーブルをつくるプログラム


```
1000 AD=&H5096
1010 FOR I=0 TO 90
1020 DAT=126*COS(I/180*3.141592654#)
1030 POKE AD,DAT:AD=AD+1
1040 NEXT I
```

プログラムの上手な作り方

株式会社 シー・シー・ダブル システムコンサルタント **湯村哲男**

●はじめに●

プログラムを上手に作るには，既存のプログラムをいかにうまくMERGEするかにある，と言っても過言ではない。そのためには自作・他作を問わず，開発済みのプログラムを豊富に持つこと，そしてそれらを必要に応じて利用できるように整理しておくことがたいせつだ。

ここではプログラム整理の仕方について説明しようと思う。

例として「多点入力による球の決定」というテーマを与えられた場合，あなたは既に図2～5のプログラムを持っていたとすれば，メインルーチン（図1）だけをつくり，図2から図5のルーチンをMERGEするだけでよい。

●プログラムの整理項目●

プログラムの整理に必要な項目として，次のようなものが考えられる。

図1　メインルーチン


```
10 REM *** ﾀﾃﾝ ﾆｭｳﾘｮｸ ﾆ ﾖﾙ ｷｭｳ (SPHERE) ***
20 ON ERROR GOTO 1000
30 CLEAR 2000
40 CLS
50 DIM X(50),Y(50),Z(50),U9(50,5),U0(4,5),J9(4)
60 LOCATE 5,1:PRINT"ﾀﾃﾝ ﾆｭｳﾘｮｸ ﾆ ﾖﾙ ｷｭｳ"
70 LOCATE 5,3:PRINT"ﾆｭｳﾘｮｸﾃﾝ ﾉ ｶｽﾞ ="
80 LOCATE 21,3:GOSUB 10000:N=VAL(E$)
90 LOCATE 5,4:PRINT"NO.     X          Y          Z"
100 L=4:FOR I=1 TO N:L=L+1:LOCATE 6,L:PRINT I
110 LOCATE 10,L:GOSUB 10000:X(I)=VAL(E$)
120 LOCATE 20,L:GOSUB 10000:Y(I)=VAL(E$)
130 LOCATE 30,L:GOSUB 10000:Z(I)=VAL(E$)
140 NEXT I
150 FOR I=1 TO N
160 U8=X(I)-X(1):V8=Y(I)-Y(1):W8=Z(I)-Z(1)
170 U9(I,1)=U8*2:U9(I,2)=V8*2:U9(I,3)=W8*2:U9(I,4)=1:U9(I,5)=U8*U8+V8*V8+W8*W8
180 NEXT I
190 I9=4:GOSUB 10100:GOSUB10200
200 X0=U0(1,5):Y0=U0(2,5):Z0=U0(3,5)
210 R0=U0(4,5)+X0*X0+Y0*Y0+Z0*Z0:R0=SQR(R0)
220 X0=X0+X(1):Y0=Y0+Y(1):Z0=Z0+Z(1)
230 CLS
240 LOCATE 10,10:PRINT"ｷｭｳ ﾉ ﾁｭｳｼﾝｻﾞﾋｮｳ"
250 LOCATE 15,11:PRINT"X =";X0
260 LOCATE 15,12:PRINT"Y =";Y0
270 LOCATE 15,13:PRINT"Z =";Z0
280 LOCATE 10,14:PRINT"ｷｭｳ ﾉ ﾊﾝｹｲ"
290 LOCATE 15,15:PRINT"R =";R0
300 LOCATE 10,20:PRINT"ﾂﾂﾞｹﾏｽｶ (Y/N) :"
310 LOCATE 25,20:GOSUB 10000
320 IF E$="Y" THEN 60
330 IF E$="N" THEN 350
340 LOCATE 25,20:PRINT SPC(20):GOTO310
350 END
```

図2　エラー処理ルーチン(ERROR)

```
1000 REM *** Error Treatment (ERROR) ***
1010 LOCATE 5,20:PRINT"Error Code NO.=";ERR
1020 LOCATE 5,21:PRINT"Error Line NO.=";ERL
1030 STOP
1040 RESUME NEXT
1050 END
```

図3　データ入力ルーチン（INPUT）

```
10000     REM  ========= INPUT SUB (INPUT) =========
10010     BLK = 1
       :  E$ = ""
       :  CAR = 0
10020     W1C$ = INKEY$
10030     IF LEN(W1C$) <> 0 THEN
            BLK = 1
       :     GO TO 10060
          ELSE
            BLK = BLK+1
       :    IF BLK = 31 THEN
              BLK = 1
10040     IF BLK < 15 THEN
            COLOR 15
          ELSE
            COLOR 7
10050     PRINT " "+CHR$(&H1D) ;
       :   GO TO 10020
10060     BLK = 1
       :  IF W1C$ = CHR$(8) THEN
              IF CAR = 0 THEN 10020
              ELSE
                CAR = CAR-1
       :        COLOR 7
       :        PRINT " "+CHR$(&H1D)+CHR$(&H1D) ;
       :        E$ = LEFT$(E$ , CAR)
       :         GO TO 10020
10070     CAR = CAR+1
       :  COLOR 7
       :  IF W1C$ = CHR$(13) THEN
            PRINT " " ;
       :    RETURN
          ELSE
            PRINT W1C$ ;
       :    E$ = E$+W1C$
       :     GO TO 10020
```

図4　正規方程式入力ルーチン（NOREQ）

```
10100     REM  *** Normal Equation (NOREQ) ***
10102     '
10104     'Equation
10106     '    a1*X1+a2*X2+・・・+ai*Xi+・・・an*Xn=X(n+1)    (1)
10108     'Normal Equation of (1)
10110     '    a1*[X1・X1]+a2*[X2・X1]+・・・+an*[Xn・X1]=[X(n+1)・X1]
10112     '    a1*[X1・X2]+a2*[X2・X2]+・・・+an*[Xn・X2]=[X(n+1)・X2]
10114     '      ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
10116     '    a1*[X1・Xn]+a2*[X2・Xn]+・・・+an*[Xn・Xn]=[X(n+1)・Xn]
10118     '                [ ] denote Summation
10120     'Input Data
10122     '    U9(j,i+1) --- Data Xi
10124     '    I9        --- n
10126     '    J9        --- Number of Experiments
10128     'Output Data
10130     '    U0(i,i+1) --- Term of Normal Equation
10132     'Note
10134     '    Following DIM Statement must be put in your Main Routine
10136     '       DIM U9(J9,I9+1),U0(I9,I9+1)
10138     '
10144     FOR J = 1 TO I9
10146       FOR I = 1 TO I9+1
10148         U0(J , I) = 0
10150       NEXT I
10152     NEXT J
10154     FOR J = 1 TO I9
10156       FOR K = 1 TO J9
10158         U1 = U9(K , J)
10160         FOR I = J TO I9+1
10162           U0(J , I) = U0(J , I)+U9(K , I)*U1
10164         NEXT I
10166       NEXT K
10168       IF J = 1 THEN 10176
10170       FOR J1 = 1 TO J-1
10172         U0(J , J1) = U0(J1 , J)
10174       NEXT J1
10176     NEXT J
10178     RETURN
```

（1）内容と機能
（2）プログラムリスト
（3）使用変数リスト
（4）使用サブルーチンリスト
（5）使用ファイルリストとその機能
（6）その他（入力変数と出力変数，使用に際

図5　正規方程式の解ルーチン(NESOL)

```
10200    REM  *** Solution of Normal Equation or Matrics (NESOL) ***
10202    '
10204    'Input Data
10206    '   U0(i,i+1) --- Term of Normal Equation ( i=1 to I9 )
10208    '   I9        --- Number of Terms of Normal Equation
10210    'Output Data
10212    '   U0(i,i+1)
10214    '     U0(i,j) (i=1 to I9,j=1 to I9) --- Inverse Matrics
10216    '     U0(i,I9+1) (i=1 to I9)        --- Solution of Normal Equation
10218    '   G9        --- =1,If Solution can't be obtained,else G9=0
10220    'Note
10222    '   Following DIMension Statement must be set in your Main Routine
10224    '       DIM U0(I9,I9+1),J9(I9)
10226    '
10228    G9 = 0
10230    FOR L0 = 1 TO I9
10232      U8 = 0
    :      L3 = 0
10234      FOR L1 = L0 TO I9
10236        U7 = ABS(U0(L1 , L0))
10238        IF U8 < U7 THEN
               U8 = U7
    :          L3 = L1
10240      NEXT L1
10242      J9(L0) = L3
10244      IF L3 = 0 THEN
             G9 = 1
    :         GO TO 10304
10246      IF L3 = L0 THEN 10258
10248      FOR J8 = 1 TO I9+1
10250        U7 = U0(L3 , J8)
10252        U0(L3 , J8) = U0(L0 , J8)
10254        U0(L0 , J8) = U7
10256      NEXT J8
10258      U8 = U0(L0 , L0)
    :      U0(L0 , L0) = 1
10260      FOR J8 = 1 TO I9+1
10262        U0(L0 , J8) = U0(L0 , J8)/U8
10264      NEXT J8
10266      FOR L1 = 1 TO I9
10268        IF L1 = L0 THEN 10280
10270        U7 = U0(L1 , L0)
10272        U0(L1 , L0) = 0
10274        FOR J8 = 1 TO I9+1
10276          U0(L1 , J8) = U0(L1 , J8)-U0(L0 , J8)*U7
10278        NEXT J8
10280      NEXT L1
10282    NEXT L0
10284    FOR L0 = 1 TO I9-1
10286      I8 = I9-L0
    :      I3 = J9(I8)
10288      IF I3 = I8 THEN 10300
10290      FOR L1 = 1 TO I9
10292        U7 = U0(L1 , I3)
10294        U0(L1 , I3) = U0(L1 , I8)
10296        U0(L1 , I8) = U7
10298      NEXT L1
10300    NEXT L0
10302    RETURN
10304    LOCATE 10 , 20
    :    PRINT "=== None Inverse Matrics ==="
10306    RETURN
```

図6

```
10000 REM ========= INPUT SUB (INPUT) =========
10010 BLK=1:E$="":CAR=0
10020 W1C$=INKEY$
10030 IF LEN(W1C$)<>0 THEN BLK=1:GOTO10060 ELSE BLK=BLK+1:IF BLK=31 THEN BLK=1
10040 IF BLK<15 THEN COLOR 15 ELSE COLOR 7
10050 PRINT " "+CHR$(&H1D);:GOTO10020
10060 BLK=1:IF W1C$=CHR$(8) THEN IF CAR=0 THEN 10020 ELSE CAR=CAR-1:COLOR 7:PRIN
T " "+CHR$(&H1D)+CHR$(&H1D);:E$=LEFT$(E$,CAR):GOTO 10020
10070 CAR=CAR+1:COLOR 7:IF W1C$=CHR$(13) THEN PRINT " ";:RETURN ELSE PRINT W1C$;
:E$=E$+W1C$:GOTO 10020
```

図7

```
I
 10146 10148 10150 10160 10162 10162 10162 10164

I9
 10144 10146 10154 10160

J
 10144 10148 10152 10154 10158 10160 10162 10162 10168 10170 10172 10172 10176

J1
 10170 10172 10172 10174

J9
 10156

K
 10156 10158 10162 10166

U0(*)
 10148 10162 10162 10172 10172

U1
 10158 10162

U9(*)
 10158 10162
```

しての諸注意など)。

●整理方法●

1.プログラムリスト

例を「データ入力ルーチン」にとって説明しよう。

BASICプログラムの特徴は，マルチステートメントでコーディング，キーイングができることだ。ところがキーインのイメージでのリスト（図6）は，構造がたいへんわかりにくい。これを図3のようにすれば，デバッギングやプログラムの改変などにとても便利になる。プログラムは図3のような形でリストにしておくことをすすめる。

2.使用変数リスト

プログラムをMERGEする場合，注意しなければならないのが使用変数だ。変数リスト作成は，プログラム整理をするにあたっての必須の条件の一つである。

プログラムをつくる段階で注意しながらリストを作ることもできるが，必ずといっていいほど漏れがあるものだ。まして他作についてのリスト作成は容易ではない。図7は図4のプログラムの使用変数リストである。

3.使用サブルーチンリスト

図1のメインルーチンによるサブルーチン

図8

```
   60 : Goto
  320

  310 : Goto
  340

  350 : Goto
  330

 1000 : Goto
   20

10000 : Gosub
   80    110    120    130    310

10100 : Gosub
  190

10200 : Gosub
  190
```

図9

```
10 REM *** トウコウセン ヲ CRT ガメン ニ エガク (DISFLY) ***
20 ON ERROR GOTO 1000
30 CLEAR
40 CLS:CLOSE#16:CLOSE#1:CLOSE#2:CLOSE#3
50 WIDTH 80,25
55 DEFSNG A-H:DEFINT I-N:DEFSTR O-R:DEFSNG S-Z
60 DIM H9(50),K5(50),IX1(1000),IY1(1000),IX2(1000),IY2(1000)
70 OPEN "O",#16,"LPTO:S"
80 OPEN"R",#1,"1:SUBDT1"
85 OPEN"R",#2,"1:SUBDT2"
90 OPEN"R",#3,"1:DRCONT"
100 GOSUB 2000
110 GOSUB 9200
120 HARDC2
130 CLOSE#1:CLOSE#2:CLOSE#3:CLOSE#16
140 RUN"CNTDRW"
150 END
1000 REM *** ERROR TREATMENT***
1010 LOCATE 5,20:PRINT"ERROR CODE NO.:";ERR
1020 LOCATE 5,21:PRINT"ERROR LINE NO.:";ERL
1030 STOP
1040 RESUME NEXT
1050 END
2000 REM ***
2002 FIELD#1,2 AS O1$,2 AS O2$,4 AS O3$,4 AS O4$,4 AS O5$,4 AS O6$,20 AS O7$,8 A
S O8$,4 AS O9$,4 AS P1$,2 AS P2$,2 AS P3$,2 AS P4$,2 AS P5$,2 AS P6$,2 AS P7$,4
AS P8$,4 AS P9$,2 AS Q0$,2 AS Q1$
2004 GET#1,1
2006 N=CVI(O1$):M=CVI(O2$):G1=CVS(O3$):G2=CVS(O4$):H1=CVS(O5$):H2=CVS(O6$):P$=O7
$:PO$=O8$
2008 BA=CVS(O9$):BI=CVS(P1$):N1=CVI(P2$):N2=CVI(P3$):N3=CVI(P4$):M1=CVI(P5$):M2=
CVI(P6$):M3=CVI(P7$):HO=CVS(P8$):DO=CVS(P9$):L3=CVI(Q0$):KOE=CVI(Q1$)
2009 N5=50+N3*(N2-N1):M5=160-M3*(M2-M1)
2010 FOR I=1 TO L3
2012 I8=INT((I-1)/42)+1
2014 I9=I-42*(I8-1)
```

リストを図8に示す。これはサブルーチンばかりでなく，GOTOなどの飛び先とそのステートメントNo.を出すリストで，デバッグなどにたいへん便利だ。

このリストを利用してサブルーチンリストを作成すれば手落ちがない。

4.使用ファイルリストとその構造

図9のプログラム中の文番号80で使用しているファイル名は“SUBDT1”で，その構造は文番号2000である。これによりSUBDT1の中で使用する変数名とバイト数がわかるから，それらとともにその変数が表す内容を具体的に書いておくとよい(図10)。

5.その他

図4，5などのREM文にある入出力変数名や注意事項などがない場合にも，それらを書きとめておくことをすすめる。

図10

ファイル・レイアウト　　（　／　）

ファイル名　SUBDT1

レコードサイズ　80

レコード数　1

A　文字列
I　整数
S　単精度実数
D　倍精度実数

レコードNo.	項目No.	データ名称	A／N	文字数	説　明	変数名
1	1	Ō1$	I	2	x軸方向Mesh Pointの数	N
	2	Ō2$	I	2	y軸方向Mesh Pointの数	M
	3	Ō3$	S	4	図面原点のx座標値	G1
	4	Ō4$	S	4	同　上 y 〃	G2
	5	Ō5$	S	4	x方向 1区間の長さ	H1
	6	Ō6$	S	4	y方向　〃	H2
	7	Ō7$	A	20		P$
	8	Ō8$				

●おわりに●

プログラムを上手につくるのに必要な条件を例をあげて説明したが，ここで使ったユーティリティは「プログラマーズ・パル」(㈱シー・シー・ダブル作成）で，FM-8用のデバッギングエイドである。プログラムのデバッグにも使えるから，このようなものを駆使してプログラムを上手に楽につくってほしい。

英文ワープロ「イージー・ライター」試用記

西中村涼子

世はまさにコンピュータ時代!! 各企業のあらゆる分野にコンピュータが導入され，マイコンの出現で，小会社や家庭への導入も可能になった。とはいえ，まだどこか業界にのせられているような感も残る。昨年は，新聞・雑誌などで「ワープロ」という言葉をよく見かけた。中にはワープロ＝コンピュータと思っている人も少なからずいる。ワープロとはワードプロセッサの略で文章作成用の機能である。ワープロ専用のコンピュータもあるし，マイコンでワープロのソフトを使って機能させることもできる。昨年のブームは，ほとんどが日本語のワープロであった。

日本だから，日本語ワープロがもてはやされて当然だが，しかし，マイコンのキーボードを見ると，何といっても英文タイプライタにそっくりで，英文を打つのが最も自然に思える。それに我々日本人の日常生活には英語という言語が深くかかわってきているし，英語を学習する者も多ければ，仕事で英語を使う者も多い。そこで，英文ワープロを取りあげてみた。

英文ワープロに要求されることは，まずシンプルさということである。ごたごたしていてはいけない。画面もシンプル，操作もシンプルに願いたい。ただし，必要な機能は備えていなければ困る。シンプルさと便利さの接点を見つけなければならない。

今回紹介するのは，FM-7，FM-8用に開発された「イージー・ライター」(ケンブリア・ランゲージ・サービス）で，以下はその試用記である。

画面は使いやすいか？

ほとんどの英文ワープロがスクロール式(画面の最下行にきたら，1行進むごとに上へ巻き上がっていく方式）を用いているのに対し，「イージー・ライター」はノートのようにページをめくっていく方式をとっている。一つの画面が1ページということで，1ページを打ち終わって，ページめくりキー PF6 を押すと，画面に2ページめが出る。全部打ち終わって前のページを見たいというときは，PF2 キーで見たいページを指示すれば，一発でアクセスできる。

スクロール方式，ページめくり方式のどちらが良いかということは，一概には言えないし好みにもよるが，両方の方式を使用してみて気付いたことは，まずスクロール式は，最下行にきたら，その後，一行ごとに上へずれていくので，いつも画面の下だけで打っていく形となり，少々窮屈さを感じた。それに対し，ページめくり方式は，普通のノート同様，伸び伸びと画面を使えるような気がする。ただし，ページをめくったとき，前のページの最後の行が表示されれば，もっと便利だろうと感じた（今後の改訂版に期待したい）。どちらも一長一短といったところだが，私の印象では，スクロール式があまりにコンピュータ的であるのに対し，ページめくり方式は，従来の文章作成の雰囲気を残していて，とっつ

★訂正前（ミスプリあり）・行末処理前★

```
   This is my house.  I live with Father, Mother and tw
o sisters.  Father is a typical Japanese businessman.
He works at a bank.  He always wanted to have a new hou
se of his own.  Father worked hard to make his dream co
me true.  Mother helped him by saving money.  Money we
spent in an averagemmonth, before we built our house, w
as
                     ①

          Rent for an apartment              ¥40,000.-
          Food                               ¥50,000.-
          Clothes                            ¥20,000.-
          Savings                            ¥50,000.-
          Others                             ¥90,000.-
                                           -----------
          In all                            ¥180,000.-
                         ③                     ②
   As I also wanted to to live in a new house, I helped
 them by delivering morning paper every day.  With this
, I earned all the money I needed for school.  Now we l
ive in a beautiful house with a small garden where we c
an grow many kinds of vegetables, so that we don't have
 to buy them at a market.  Money we spend now is

          House loan                         ¥70,000.-
          Food                               ¥35,000.-
          Clothes                            ¥20,000.-
          Savings                            ¥20,000.-
          Others                            ¥120,000.-
                                           -----------
          In all                            ¥265,000.-
                                                    ④
   I expected that Father would say to me, "Kaz, you ha
ve to deliver newspaper any more."  He, however, said,
"Kaz, how about delivering evening paper, too?", instea
d.
```

――――……行末で単語が切れてしまっている

①………不要文字mがはいっている　②………数値が間違っている

③………不要語toがはいっている　④………don'tが抜けている

きやすいと思った。また何ページのどこらへんに何を書いたという視覚的な記憶も捨てがたい。

画面のデザインは，シンプルで必要最低限の表示にとどめている。ファンクションキーの説明表示がどうしても欲しいときは[PF1]キーを押せば，画面上部に1行だけで表示される。

修正は簡単か？

さて，シンプルなだけではワープロの意味がない。文章作成をスムーズに行うため，どんな修正も速やかにこなしたい。従来のタイプライタで英文を作成するのは，誠に面倒で仕上げに手間どる。99％間違いがないのに，完璧な仕上がりにするため最初から打ち直さねばならなくなることがある。余計な時間と労力を費やし，しかも精神的にいらいらする。こんな煩雑なタイプ作業を経験した人にとって，画面上で文字を上下左右に移動させたり，消去したりできるプログラムがあるなんて，なんとありがたいことだろう。

キー操作もいたって簡単。一例を挙げると，単語や文を挿入したいという場合，一文字だけの挿入なら[INS]キーを，たくさんの挿入なら[PF4]キーでその数を指定すれば，カーソルの所からその数だけ空白をつくってくれる。削除の場合も，[DEL]キーと[PF5]キーで同じように操作すれば，その分だけ前につめて不要部分を削ってくれる。処理スピードは，BASICで書かれているにもかかわらず，全く気にならない。

このように簡単にスピーディに修正できるので翻訳のときなど，下書きをする必要がなくなった。英文を作成しようと思ったら，すぐコンピュータに向かえばよい。

行末の処理は？

タイピングでもう一つ面倒なことがある。それは行末の処理である。従来のタイプライタの場合，行が終わりかけたら，次の単語が入りきるかどうか，長い単語ならどこで区切ればよいかと，いちいち考えなければならない。ところが「イージー・ライター」を使うと，その心配は全くいらない。行末など気にせず，ひたすら打ち続ければよい。すべての行が後から後からつながって，1行を形成しており，画面の都合上，同幅で切って並べただけだと考えればよい。たとえ，画面の行末で単語が切れてしまっても，コンピュータは正しい文を読んでくれている。すべてを打ち終わり，修正を終えた段階で[PF7]キーを押せば，画面上で単語がきれいに整頓されていく。行末で切れてしまっている単語も，スペースの計算により，上か下の行にうまくはめこまれていく。この行末処理，一瞬で完了！というわけにはいかないが，両端がきれいにそろった美しい英文ができあがる。

★訂正後・行末処理後★

```
   This is my house.  I live with  Father,  Mother  and
two sisters.  Father is a typical Japanese businessman.
He  works  at  a  bank.  He always wanted to have a new
house of his own.  Father worked hard to make his dream
come true.  Mother helped him by saving money.   Money
we  spent  in  an  average  month,  before we built our
house, was

      Rent for an apartment               ¥40,000.-
      Food                                ¥50,000.-
      Clothes                             ¥20,000.-
      Savings                             ¥50,000.-
      Others                              ¥90,000.-
                                        -------------
      In all                             ¥250,000.-

   As I also wanted to live in a new  house,  I  helped
them by delivering morning paper every day.  With this,
I  earned  all  the  money I needed for school.  Now we
live in a beautiful house with a small garden where  we
can  grow  many  kinds  of vegetables, so that we don't
have to buy them at a market.  Money we spend now is

      House loan                          ¥70,000.-
      Food                                ¥35,000.-
      Clothes                             ¥20,000.-
      Savings                             ¥20,000.-
      Others                             ¥120,000.-
                                        -------------
      In all                             ¥265,000.-

   I expected that Father would say to  me,  "Kaz,  you
don't  have  to  deliver  newspaper  any  more."   He,
however,  said,  "Kaz,  how  about  delivering  evening
paper, too?", instead.
```

どう保存するか？

まず，プリンタにつないで印字することができる。最終的に作成するペーパーの大きさで，打った英文が何枚分になるかということは，最初にマージン（1行の文字数）を指定するとき，目安表が出てくる。例えば，マージンを79に決めて70行（つまり画面3ページ半）打てば，B4サイズの1枚分になるといった具合だ。プリンタは，ほとんどの機種が使える。コンピュータ文字が気に入らないという人は，電動タイプライタ用のソフトもあると言う。

英文を紙面に残すだけなら，後で修正を施すといったことはできないが，カセットテープに記録しておけば，いつでもその英文を呼び出して修正することが可能だ。これもコンピュータならではの業で，英文の保管・整理がやりやすくなった。

＊　＊　＊

英文ワープロは，今までのタイプライタとは比較にならないくらい便利なものだ。翻訳も手軽にできるし，実験データ記録などもスムーズにできる。また，タブセットもできるし，カナモードでも使えるので，テキストやテストの作成にも使える。テープ保管ができるので，2，3年前のテストをタイトルだけかえて再び使用することもできる。

プロばかりではない。一般の学生にも重宝な品だ。タイプの苦手な人も，レポート提出のときにあわてることもない。また，英語の小説などをコピーしてみるのもいい勉強になるだろう。

イージー・ライターに関するお問い合わせは，最寄りの日本ソフトバンク加盟店か，ケンブリア・ランゲージ・サービス（〒890 鹿児島市鴨池1-11-21 キクビル6F ☎0992-52-2108）まで

for FM-7, FM-8

タネ明かし懸賞付き アニマルミステリー

テンヨー 加藤 英夫

前回のカードテレパシーではコンピュータが超能力者であり，コンピュータがカードを当てる，というプログラムであったが，今回のアニマルミステリーは，超能力者はキミでコンピュータを道具に使った手品だ。もちろんキミが超能力者であるはずはなく，当然タネがある。このタネを，つまりコンピュータとキミがいかにしてコミュニケートするのかを，見破ってほしい。

遊び方

このプログラムは相手に好きな動物を選ばせ，それをテレパシーで送ってもらい，それを当てるというものだ。

初めにCRTに

ウ シ

という動物名が表示される。そこでキミは，

「このようにPF1キーを押していって好きな動物の名前が出たところで手を止めてください。」と説明しながら最後にPF2キーをなにげなく押してしまう(キーポイント1)。

「それでは始めましょう。私は後ろを向いていますから，どんどんキーを押して動物を選んでください。」

「決まりましたか？　それではその動物の名前をテレパシーで私の方へ送ってください。」と言ってキミは一生懸命テレパシーを受け止める演技をする。この演技は少なくとも，1から5までをゆっくり数えるまでは続けなければならない(キーポイント2)。

「あなたの考えている動物が，だんだんはっきりした像になってきました。それではもう一度PF1キーを何度か押して，あなたが選んだ動物名を隠してください。」

「隠し終わりましたか？　それではあなたがテレパシーで送ってくれた動物を当ててみましょう。」

ここでキミはコンピュータに向き直り，PF1キーを何回か押しながら，

「キミは何回でも自由にキーを押したのだから，キミの選んだ動物を当てるのは不可能ですよね。」

などと言いながら，1回だけこっそりPF2キーを押す(キーポイント3)。そのとき画面に出た動物名が答えであるから，それをしっかり記憶する。PF2キーを押したあと，あと1，2回PF1キーを押して答えの動物名を画面から消すのをお忘れなく。

あとはいかにも精神集中をするような演技をしたあと「あなたの選んだ動物は○○○です。」と言って当ててもよいし，PF1キーを押し続けて答えの動物がもう一度出てきたところで手を止めて，「あなたがテレパシーで送ってくれたのはこの動物ですね。」と言って終わってもよい。

さて，キミの使命だが……

このゲームのキーポイントは，遊び方で示したように3つある。このうち最も重要なポイントがキーポイント2，キーポイント3である。

それではプログラムを見てキーポイント2，3を解いてみよう。そしてこの超能力者の化けの皮をはがしてほしい。

- **キーポイント1**　“これから相手がキーインしますよ”というシグナルをコンピュータに伝えるものだ。
- **キーポイント2**

キーインのサブルーチンの中でインクリメントした変数T%が，行番号1400で定数[　　]より大きいかどうかチェックし，大きい場合には変数[　]に選ばれた動物の番号が記憶される。

- **キーポイント3**

行番号[　　]でPF[　]キーが再び押されたと判断したら，行番号1420で答えの動物名を画面に表示する。

応募要領

- 上の[　　]の中にはいる数字や文字を官製はがきに書いて送ってください。
- 住所・氏名・年齢・職業・電話番号も忘れずに。
- 4月30日の消印まで有効です。
- 正解者の中から抽選で50名の方に，Oh！FM Tシャツをプレゼントいたします。
- 宛先は
〒102　東京都千代田区四番町3
四番町ハイツ501 ㈱日本ソフトバンク
「Oh！FMアニマルミステリー」係

```
1000 ''''''''''''''''''''''''''''''''''''''''
1010 '                                      '
1020 '     COMPUTER MAGIC 「ｱﾆﾏﾙ ﾐｽﾃﾘｰ」     '
1030 '         - WITHOUT KANIJI ROM-        '
1040 '   CREATED & PROGRAMMED BY H.KATO     '
1050 '         (C) JANUARY 25,1983          '
1060 '                                      '
1070 ''''''''''''''''''''''''''''''''''''''''
1080 DIM AN$(20):LIMIT=500（FM-7の場合はクロックがFM-8と同じになるようにディップスイッチを変えてください。）
1090 KEY1,"A":KEY2,"B"
1100 CLS
1110 SYMBOL(80,20),"ANIMAL MYSTERY",4,4,2
1120 COLOR 7
1130 RESTORE 1160:FOR I=1 TO 20:READ AN$(I):NEXT
1140 GOTO 1260
1150 ''' ﾄﾞｳﾌﾞﾂ ﾃﾞｰﾀ '''''''''''''''''''''''
1160 DATA ｳ ｼ,ｲ ﾇ,ｳｻｷﾞ,ｷﾂﾈ,ｿﾞｳ,ｷﾘﾝ,ﾌﾞﾀ,ﾀﾇｷ,ﾈ ｺ,ｽｽﾞﾒ,ﾄ ﾗ,ｳ ﾏ,ﾂ ﾙ,ﾔｷﾞ,ﾍﾋﾞ,ｻ ﾙ,ﾘ ｽ,
ｶ ﾒ,ｼ ｶ,ﾗｲｵﾝ
1170 ''' ｷｰ ｲﾝ''''''''''''''''''''''''''''''
1180 T%=0
1190 A$=INKEY$:T%=T%+1:IF A$="" THEN 1190
1200 IF A$<>"A" AND A$<>"B" AND A$<>" " THEN 1190
1210 RETURN
1220 ''' ﾌﾟﾘﾝﾄ ''''''''''''''''''''''''''
1230 LOCATE 18,10:PRINT"     ":LOCATE 18,10:PRINT AN$(I)
1240 RETURN
1250 ''' ﾒｲﾝ ﾙｰﾁﾝ '''''''''''''''''''''''
1260 LINE@(259,89)-(376,117),PSET,1,B:LINE@(268,94)-(367,112),PSET,5,B
1270 I=1
1280 GOSUB 1230:'ﾌﾟﾘﾝﾄ
1290 GOSUB 1180:'ｷｰ ｲﾝ
1300 IF A$<>"A" AND A$<>"B" THEN 1290
1310 GOSUB 1450:'ﾜｲﾝﾄﾞ ｱｯﾌﾟ
1320 GOSUB 1230:'ﾌﾟﾘﾝﾄ
1330 IF A$<>"B" THEN 1290
1340 GOSUB 1180:'ｷｰ ｲﾝ
1350 GOSUB 1450:'ﾜｲﾝﾄﾞ ｱｯﾌﾟ
1360 GOSUB 1230:'ﾌﾟﾘﾝﾄ
1370 X=0
1380 GOSUB 1180:'ｷｰ ｲﾝ
1390 IF A$="B" THEN I=X-1:GOTO 1410 ELSE IF A$=" " THEN 1100
1400 IF T%>LIMIT THEN IF X=0 THEN X=I
1410 GOSUB 1450:'ﾜｲﾝﾄﾞ ｱｯﾌﾟ
1420 GOSUB 1230:'ﾌﾟﾘﾝﾄ
1430 GOTO 1380
1440 ''' ﾜｲﾝﾄﾞ ｱｯﾌﾟ '''''''''''''''''''''
1450 I=I+1:IF I>20 THEN I=1
1460 RETURN
```

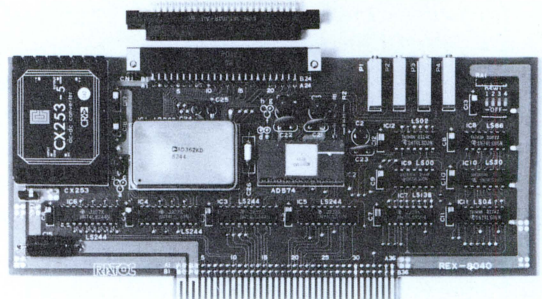
新製品情報　A/D変換カードREX-8040

ラトック・システム・エンジニアリング㈱（大阪市西区江之子島1-6-2奥内8号ビル ☎06(445)1657）では，FM-7，FM-8 の周辺機器として，12bitのA/D変換カードREX-8040 の販売を開始した。

これは，同社のI/O拡張ユニットREX-8 のオプションカードとして既に発売されているPIAカード（REX-8010），電子天秤インタフェースカード（REX-8030），GPIBカード（REX-8020）に続いて発売されたもので，定価は198,000円。同社では引き続き，D/A変換カード，シリアルインタフェースカードの発売も予定している。 主な仕様は次のとおり。

●チャンネル数：16（シングルエンド）/8（ディファレンシャル）モード切り換えはジャンパによるハードウェア設定 ●入力レンジ：0～10V，−10～+10V，−5～+5V ●A/D解像度：12ビット ●直線性誤差：±2LSB以内 ●変換時間：35μs ●カードアドレス：FDX0～FDX2（ディップスイッチでX＝8～F を設定） ●基板寸法：220×101mm ●消費電力：1,000mA×5V以下 ●コネクタ：富士通FCN-361P/361J-048

A/Dコンバータ

READER'S AREA

▶Oh！シリーズがPCとMZだけだったので，FMユーザーといたしましては，FM-8はマイナーなのではないかと思い，くやしく思っておりました。そして，やっと出ました。ワクワクして家でページをめくると，FM-7のショックが大きかっただけだった。(千葉市・阿部明宏・25)

▶やっとFM-8も貴社に認められ，その喜びは何より大きなものです。PCやMZのユーザーから，もうバカにされないですむでしょう。今後は，なるべく早く月刊誌となるよう，祈っています。(文京区・萩原忠彦・23)

▶FMシリーズ専門の雑誌が創刊されて，大変うれしく思っています。以前から，Oh！HC，MZはあったのになぜFMはないのかと思っていましたが，FM-7の参考になる記事をこれからどんどん頼みます。(京都府・大畑浩樹・21)

▶今，相当"イカリ"に燃えています。なんでFM-7なんか出たんや‼FM-8の存在価値なんかあらへんやないか‼(大阪市・松本治・22)

▶いいですね。やっと出たOh！FM，期待どおりです。もう少しグラフィック関係の記事が欲しいです。(金沢市・寺田隆・21)

▶やっとOh！FMが出ましたね。私はマイコンは持っていませんが，FMシリーズのファンです。ところでなぜ月刊誌として発刊しなかったのですか。1日も早く月刊誌としてください。(町田市・武藤一弘・18)

▶初心者の私でも楽しく読めました。この本を読んでついにFM-7を買うことに決定しました。このうえは早くOh！FMが月刊となることを期待します。(横浜市・石田芳朗・20)

――大勢の読者から早く隔月刊，月刊誌に，というお便りをいただき，ありがとうございました。第3号よりいよいよ隔月刊になります。このうえは早く月刊誌，週刊誌(？)になるよう努めますので，みなさんも応援してください。

▶もうすぐFM-7を購入したいと思っているので，今回のFM-7の特集はとてもよかった。何よりもOh！FM創刊バンザイ！――数か月後，富士通ユーザーになる人間より――(群馬県安中市・山崎敬之・16)

▶「Oh！PC」,「Oh！MZ」が出て,「Oh！FM」が出るのを待ちわびていました。記事がFM専門なので，他のマイコン誌より利用価値が高くとてもうれしい。できたら，季刊より隔月刊になってくれるとありがたい。(長野市・藤井雅光・20)

▶とてもうれしい！早く月刊誌になってほしい。ただLISTがかすれて読みづらい。(中野区・米田育・19)

――ジミー・ラミーのプログラムリストが不鮮明というおしかりを多くの方々から受けました。ごめんなさい。124ページに再掲載しました。

▶出版おめでとうございます。FMのユーザーにとってとてもうれしいことです。内容もすっきりしていて，実にいい雑誌だと思います。ただ，ぜいたくを言うようですが，ところどころに誤字やミスがめだつようです。リストなど大事なところが少し見えにくいので，その辺を改めていけばもっといい雑誌になるのではないでしょうか。(青梅市・小山田幸生・14)

▶おもしろい，ためになる，目が悪くなる。もっとゲームのプログラムリストを載せてほしかった。おもしろいゲームを作ってほしい。(塩尻市・金子敏雄・17)

▶FM-8に関する記事についてのみ言えば，見飽きたものばかりという気がする。俺としてはFM-8をマシン語レベルで動かすためのサブルーチン集とか，ゲーム作りテクニック(マシン語編)とかをのせてほしい。創刊号ということで，これからの発展に期待する。(杉並区・亀井寛・18)

▶FM-7を購入したばかりの私にはOh！FM創刊はグッドタイミングです。入門者にとっては多少，むつかしい内容もありますが，入門者向けの記事も増やしてください。紙質，印刷ともよく，長期に渡りFM-7の辞書として利用させてもらいます。(宇治市・向上茂樹・31)

▶うっ，嬉しい‼感動的‼絶句……。書店で悶絶しそうになった。Oh！FMやっと出てくれたのねえ―ぐすん，ぐすん。

(松戸市・篠原篤・21)

――篠原さん，可愛いイラストありがとう。ちょっとはにかんだ女の子に，おもわずほほえんでしまいました。イラストが趣味という篠原さん，また送ってください。このコーナーでは読者の方のイラストなども募集しております。ぜひご応募ください。

▶FM-8は忘れられていなかった‼と感激した。T嬢へ，鼻毛が伸びてもよいから月刊にしなさい。(福山市・柏原和宜・29)

▶とうとうでたか，と思わず買ってしまいましたが，できれば受験の最後の追い込みの時期に創刊してほしくなかった。徹底比較で，基板の写真までのっていたのがたいへんよかった。(江東区・秋元克紀・17)

▶前略。FM-8に関しての情報がなかなか入らず，小生も困っておりました。そこへこのOh！FMを発刊していただき，本当に有難く読ませていただきました。これでFM-8を有効に使いこなしていけるのでないかと，毎晩眠い目をこすりながら，パソコンと戦っております。

最後にひとつ，スタッフのT嬢様。鼻毛が伸びた由，それは鼻毛ではなく鼻の下が伸びた

のではないでしょうか。毎日，パソコン野郎にかこまれて。これも医学的根拠はありませんが。(結城市・杉山勉)

——鼻毛の件に関し，二，三ご考察をいただき，誠に有難く感じております。この件の真偽ですが，それは闇に葬るとして，杉山氏の考察は誠に含蓄が深いものです。パソコン野郎にかこまれると鼻の下が伸びるのだろうか。２月14日を機に検討しましたが，答えは見つからず。編集部に関して言えばノーコメントにしておいた方がよいのでは，と思いますが……(Ｔ)

▶新機種FM-７，11が発売されたのでズームアップされた内容ですが，もっとFM-８の記事も多くのせてほしいです。またFM-８のシステムその他の改造についても記載してほしかった。(大阪市・井上孝雄・29)

▶「Oh！PC」「Oh！MZ」が前から出ていてなぜ「Oh！FM」が出ないのか、と思っていたら書店で見つけたのでさっそく買ってきた。FMシリーズだけの記事なのでとてもよい。(前橋市・結城透・14)

▶FMシリーズの専門誌であるというので期待していたが，それほどでもなかった。連載記事というのはどうも読みにくい。(松原市・新居光広・33)

▶「マシン語入門」は専門用語が多いので，初心者にはわかりにくい。もっとわかりやすくしてほしい。(川西市・牧田直樹・17)

▶各雑誌に載っているFM関係の記事を集めただけという感じ。(横須賀市・渡辺博之・21)

▶価格の割に内容がうすい。(明石市・村上浩一・18)

▶ツギハギだらけであまり中身のない雑誌になってしまっているようで，ちょっと失望してしまいました。(盛岡市・佐野公昭・23)

▶もうすこし詳しい記事を期待する。目次をもっと詳しく，広告目次もつけて。(常陸太田市・綿引猛・35)

▶創刊号P.35のFM-７ベンチマークに関して『CIRCLE，PAINTを使用すると比率が高くなるのは新アルゴリズムだから』という記述ですが，わが倍速クロック(メインもサブも)FM-８の同ベンチマークの結果は，

	FM-7	FM-8	倍速FM-8	比	サブのみ倍(sec)
Pro.1	164	273	158 −6	0.96	275
Pro.2	275	449	275 ±0	1.00	275
Pro.3	38	70	40 +2	1.05	43
Pro.4	22	45	22 ±0	1.00	22
合計	499	837	493	0.99	615

ということで，ほぼFM-７と同じ結果を得ます。FM-７のためにF-BASICを書きかえたとは思えません(横浜市・中川垣明)

——FM-７ベンチマークについては他にも数名の方々から同様なご指摘がありましたので，ここでもう一度FM-７と８のスピードの違いについて触れたいと思います。まず，クロックの原発振が異なるわけです。FM-８のメインCPUの原発振は，4.9MHz，サブCPUは４MHzとなっています。一方FM-７はメイン，サブとも同一クロックで原発振は８MHzと，メイン4.9MHz，サブ４MHzの切換え式となっています。原発振とはクロックの基準となる水晶の発振周波数のことで，実際のCPUクロックはこの¼の周波数です。このように基準クロックを比べた場合FM-７を８MHzにすれば，単純計算でメインは約1.6倍，サブは約２倍，FM-８より速くなります。つまり処理速度はそれぞれのCPUの分担する割合により1.6～2倍速くなります。以上はアルゴリズムがない場合です。

さてベンチマークテストの結果からアルゴリズムの変更があったのか考察してみましょう。プログラム１はほとんどメインCPUの処理で，速度比もクロック比とほぼ等しくなっています。プログラム４はほとんどサブCPUの処理で，これも速度比とクロック比はほぼ等しくなっている，つまりPAINTのアルゴリズムに変更はなかったものと考えられます。中川氏のデータもこれを裏付けています。プログラム３はサブCPUが主なように思われますが，FM-８のCIRCLEは64角形の多角形として処理され，サブCPUにはLINEのデータが送られます。するとメインの負担が結構大きくなり，速度比がメインCPUのクロック比とサブCPUのクロック比の間となる事実と符号します。また，FM-８特有の円の描き方および多少ごつごつした円の形状は，FM-７でもそのままです。つまりアルゴリズムの改良はなかったといっていいでしょう。

ただメインCPUの変更はあるようで，FM-７とFM-８を同一クロックにした場合，FM-７の方がわずかに速くなっています。

比がぴったり一致しない点についてですが，TIME$による測定では，1秒未満の誤差を含むこと，マニュアルに記載されれているクロック周波数が概数で四捨五入されている(4.9MHzは正確には，4.9152MHzなど)ことなどが考えられます。

特に前者はタイミング次第では同一の機械でも結果が異なる原因となります。また，BASICは，BASICの変更による影響もあるようです。

以上読者からのご指摘をもとに再検討し，前号のベンチマークテストの補足とさせていただきます。

比較に用いたFM-８はバージョン1.05です。

	FM-7 (8MHz)	FM-7 (メイン4.9MHz サブ4.0MHz)	FM-8
Pro.1	164	273	275
Pro.2 LINE	274	448	449
Pro.3 CIRCLE	38	66	69
Pro.4 PAINT	22	45	45

＊すべてDISKsystem(FM-８はV1.0)

▶発刊を渇望している間に新機種が発売されてしまい，若干のあせりを感じています。なかなかユニークな編集を好感をもって迎えました。早期に隔月刊行を期待します。(大分市・兼田太郎・57)

▶富士通の新機種FM-７の内容が詳しく知りたかったので，購読した。各記事において細かな矛盾はあるもののかなり参考になった。ようやく貧乏学生にも手の出せそうな価格に興奮させられている。ところでGF募集中！ザブングル，ゴーショーグン，ミンキーモモ，イデオン，その他なんでもファンの人，お便り待ってます！(大田区・内山登志夫・18)

Information

▶このコーナーのために幅広いご意見・ご感想・ご要望などをお寄せください。内容は問いません。イラストなども歓迎です。

アンケートハガキに書き足らない場合は，官製ハガキか封書を利用してください。掲載された方にはOh！FMステッカーをさしあげます。あて先は

〒102 千代田区四番町３　４番町ハイツ501

日本ソフトバンク「Oh！FM」エリア係

▶FMに関する質問に次号より誌上で答えます。質問内容を文書でお寄せください。住所・氏名・TELを明記して，質問係あて送ってください。

▶30行以内のプログラムを募集します。楽しい画面プログラム，便利なサブルーチンなど内容は問いません。ただし，マルチステートメントはあまり多用しないでください。あて先は，Oh！FMサブルーチン集係まで。

▶第３号は５月18日(水)発売予定です。(隔月刊)

虫つくろいのページ

（創刊号の訂正）

色の部分が訂正事項です。

P.35　リスト1

170　PRIME　　I ＋ I ＋ 3

⇓

170　PRIME ＝ I ＋ I ＋ 3

P.57　I/Oアドレス表

FD03のb0　スピーカー　0：ON　1：OFF

⇓

0：OFF　1：ON

P.64　〔別表1〕6809実行命令一覧表

BEQの動作　Branch　C ＝ 1

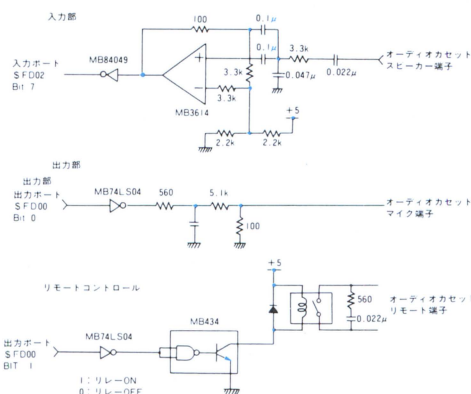
⇓

Branch　Z ＝ 1

P.56　図5　オーディオカセット入出力インターフェース部回路図

入力部
入力ポート SFD02 Bit 7
MB84049
MB3614
100
0.1μ
0.1μ
3.3k
3.3k
3.3k
0.047μ
0.022μ
オーディオカセット スピーカー端子
+5
2.2k
2.2k

出力部
出力部
出力ポート SFD00 Bit 0
MB74LS04
560
5.1k
100
オーディオカセット マイク端子

リモートコントロール
+5
MB434
560
0.022μ
オーディオカセット リモート端子
出力ポート SFD00 BIT 1
MB74LS04
1：リレーON
0：リレーOFF

P.63　プログラム5

```
         LDA    #$00
         STA    #$7000   → $7000   $7000…和が入る
         LDA    #$01
LOOP     STA    $7001    $7000 → $7001…カウンター
                         この値を1つずつふやして
         ADDA   $7000 → $7001   $7000 → $7001に加えていく
         STA    $7000
         LDA    $7001
         ADDA   #$01
         CMPA   #$0B     1を足して11(＝$0B)になったら終わり
         BEQ → BNE  LOOP
         RTS
```

ジミーラミー・プログラムリスト（不鮮明なため再掲）

```
0 ' ♠♥♦♣♠♥♦♣♠♥♦♣♠♥♦♣♠♥♦♣♠♥♦♣♠♥♦♣♠♥♦♣♠
1 ' ♠♥♦♣                              ♥♦♣♠
2 ' ♠♥♦♣  J I M M Y   R A M M Y       ♥♦♣♠
3 ' ♠♥♦♣                              ♥♦♣♠
4 ' ♠♥♦♣ nichikon-sekkei              ♦♣♠♥
5 ' ♠♥♦♣                              ♥♦♣♠
6 ' ♠♥♦♣♠♥♦♣♠♥♦♣♠♥♦♣♠♥♦♣♠♥♦♣♠♥♦♣♠♥♦♣♠
7 '
10 WIDTH80,25:DEFINT A-Z
20 DIM CARD(52),AJ(35),AQ(35),AK(35),AJO(35),PC(2,9),DC(22),PN(2,9),PS(2,9),CN(2
,9),CS(2,9),TN(9),TS(9),TC(22),LT(9),WN(9),WS(9),SC(2,11),SCS(2,11),SCN(2,11),MN
E(2)
30 FOR K=0 TO 2:MNE(K)=5000:NEXT
40 SYMBOL (130,0),"Jimmy Rammy",4,2,4
50 SYMBOL (130,170)," ニ チ コ ン ",3,2,5
60 SYMBOL (120,70),"Initializing",4,4,6
70 FOR I=0 TO 52:CARD(I)=I:NEXT
80 RESTORE 3490:FOR I=0 TO 35:READ AJ(I):NEXT
90 FOR I=0 TO 35:READ AQ(I):NEXT
100 FOR I=0 TO 35:READ AK(I):NEXT
110 FOR I=0 TO 35:READ AJO(I):NEXT
120 X=120:Y=20:FOR W1=0 TO 52 STEP 4:W=INT(RND(1)*4)+W1:GOSUB 2480:GOSUB 2650
130 X=X+53:IF X=491 AND Y=20 THEN X=120:Y=120:NEXT W1
140 NEXT W1
150 FOR K1=0 TO 5:RANDOMIZE(TIME/3):I=INT(RND(1)*52)
160 FOR K=I TO 52:J=INT(RND(1)*52):SWAP CARD(K),CARD(J):NEXT K,K1
170 SYMBOL (120,70),"Initializing",4,4,6,,XOR
180 GOSUB 1670:FOR ME=0 TO 1000:NEXT
190 CLS:GOSUB 2390:GOSUB 1600
200 FOR I=0 TO 1:GOSUB 240:NEXT
210 GOSUB 420
220 GOTO 200
230 COLOR7:END
240 COLOR(I+1)*2:LOCATE0,22:PRINT "Player";I+1;"'s turn, thinking !";
250 E=E(I):IF TC(DT-1)=52 THEN 290
260 IF E<2 THEN EN=0:TES=0:GOSUB 1500
270 IF EN=1 THEN EN=0:TES=0:RETURN
280 GOSUB 2000:IF A<2 AND B<2 THEN 290 ELSE 990
290 FOR J=E TO 0 STEP -1:IF CS(I,J)=0 AND CN(I,J)=0 THEN 320 ELSE NEXT J
300 FOR J=E TO 0 STEP -1:IF CS(I,J)=0 AND PC(I,J)<>52 THEN 320 ELSE NEXT J
310 FOR J=E TO 0 STEP -1:IF PC(I,J)<>52 THEN 320 ELSE NEXT J
320 SWAP PC(I,J),TC(DT-1):PA=PC(I,J):JJ=J:GOSUB 2530:GOSUB 2000
330 IF A<2 AND B<2 THEN 340 ELSE J=JJ:GOSUB 2540:GOSUB 2580:W=TC(DT-1):GOSUB 252
0:RETURN
340 IF PA<>52 THEN 350 ELSE J=JJ:GOSUB 2540:GOSUB 2580:W=TC(DT-1):GOSUB 2520:RET
URN
```

```
350 FOR J=0 TO E:IF PC(I,J)=PA THEN SWAP PC(I,J),TC(DT-1):NEXTJ ELSE NEXT J
360 FOR J=E TO 0 STEP -1:IF CS(I,J)=0 AND CN(I,J)=0 THEN 390 ELSE NEXT J
370 FOR J=E TO 0 STEP -1:IF CS(I,J)=0 AND PC(I,J)<>52 THEN 390 ELSE NEXT J
380 FOR J=E TO 0 STEP -1:IF PC(I,J)<>52 THEN 390 ELSE NEXT J
390 GOSUB 2540:GOSUB 2550:W=PC(I,J):GOSUB 2520:TC(DT)=W:DT=DT+1
400 GOSUB 2560:PC(I,J)=DC(DN):JJ=J:DN=DN+1:IF DN>22 OR DT>22 THEN GOSUB 1570
410 GOSUB 2530:J=JJ:GOSUB 2540:GOSUB 2580:RETURN
420 I=2:E=E(I):GOSUB 2570
430 LOCATE 0,22:COLOR 5:PRINT 'Your turn next. COMMAND: ';:A$=INPUT$(1):PRINTA$
440 IF ASC(A$)>47 AND ASC(A$)<49+E THEN 480
450 IF ASC(A$)=13 THEN 570
460 IF ASC(A$)=11 THEN 840
470 IF ASC(A$)=18 THEN 730 ELSE GOSUB 2570:GO TO 430
480 J=VAL(A$):W=PC(I,J):GOSUB 2540:GOSUB 2550:GOSUB 2570
490 LOCATE 0,22:COLOR 5:PRINT 'Draw new card (D) or Thrown card (T) ';:A$=INPUT$
(1):PRINTA$
500 IF ASC(A$)=68 THEN 520
510 IF ASC(A$)=84 THEN 540 ELSE GOSUB 2570:GO TO 490
520 GOSUB 2520:TC(DT)=W:DT=DT+1:W=DC(DN):DN=DN+1:IF DN>22 OR DT>22 THEN GOSUB 15
70
530 GOSUB 2560:GO TO 550
540 GOSUB 2520:SWAP W,TC(DT-1)
550 PC(I,J)=W:GOSUB 2540:GOSUB 2480:GOSUB 2650
560 GOSUB 2530:FOR J=0 TO E:W=PC(I,J):GOSUB 2540:GOSUB 2480:GOSUB 2650:NEXT J:GO
SUB 2570:RETURN
570 GOSUB 2570:LOCATE 0,22:COLOR 5:INPUT 'Input card numbers (xyz) ',A$
580 IF LEFT$(A$,1)='E' THEN GOSUB 2570:GO TO 420
590 LE=LEN(A$):IF LE<3 THEN GOTO 570
600 FOR J=1 TO LE:LT(J)=VAL(MID$(A$,J,1)):IF LT(J)>E THEN GOSUB 2570:GO TO 570 E
LSE NEXT J
610 FOR J2=1 TO LE-1:FOR J=1 TO LE-1:IF LT(J)>LT(J+1) THEN SWAP LT(J),LT(J+1):NE
XTJ,J2 ELSE NEXTJ,J2
620 FOR J=1 TO LE:W=PC(I,LT(J)):GOSUB 2510:WN(J)=N:WS(J)=S:NEXT J
630 IF WN(LE)=14 THEN LEF=LE-1 ELSE LEF=LE
640 FOR J=1 TO LEF:IF WN(J)=WN(1) THEN NEXT J:GO TO 690
650 FOR J=1 TO LEF:IF WS(J)<>WS(1) THEN 570 ELSE NEXT J
660 IF WN(LE)=14 THEN FOR J=1 TO LEF-1:IF WN(J+1)-WN(J)=2 THEN WN(LE)=WN(J)+1:GO
 TO 670 ELSE NEXT J:GO TO 680
670 LEF=LE:FOR J2=1 TO LEF-2:FOR J=1 TO LE-1:IF WN(J)>WN(J+1) THEN SWAP WN(J),WN
(J+1):NEXT J,J2 ELSE NEXT J,J2
680 FOR J=2 TO LEF:IF WN(J)-WN(J-1)<>1 THEN 570 ELSE NEXT J
690 FOR J=LE TO 1 STEP-1:X=(LT(J))*53:Y=(I+1)*40:LTJ=LT(J):GOSUB 2550:XD(I)=XD(I
)+1:SC(I,XD(I))=PC(I,LT(J)):PC(I,LTJ)=60:NEXT J:FOR J=0 TO E:GOSUB 2470:NEXT J:E
(I)=E(I)-LE:E=E(I)
700 GOSUB 2530:GOSUB 2210:FOR J=0 TO E:W=PC(I,J):GOSUB 2480:GOSUB 2540:GOSUB 265
0:NEXT J:FOR J=0 TO E:GOSUB 2470:NEXT
710 FOR J=0 TO XD(I):W=SC(I,J):GOSUB 2480:X=583-53*J:Y=(I+1)*40:GOSUB 2650:NEXT
J
720 GOSUB 2570:IF E<0 THEN 1770 ELSE RETURN
730 GOSUB 2570:IF E>1 THEN 420
740 LOCATE 0,22:COLOR 5:PRINT 'Input card number : ';:A$=INPUT$(1):PRINT A$
750 IF LEFT$(A$,1)='E' THEN GOSUB 2570:GO TO 420
760 IF VAL(A$)<0 OR VAL(A$)>E THEN 730
770 GOSUB 2570:LOCATE 0,22:COLOR 5:PRINT 'Input PLAYER's number (1,2,3 from top)
: ';:P$=INPUT$(1):PRINT P$
780 IF VAL(P$)<1 OR VAL(P$)>3 THEN 770
790 SW=VAL(A$):W=PC(I,SW):GOSUB 2510:SWN=N:SWS=S:K=VAL(P$)-1:GOSUB 1080
800 IF EN<>1 THEN GOSUB 2570:LOCATE 0,22:PRINT 'Cannot insert card !':FOR SOL=0
TO 500:NEXT SOL:GOSUB 2570:GO TO 420
810 GOSUB 2570:IF SW=1 THEN J=1:GOSUB 2470:RETURN
820 J=1:GOSUB 2540:GOSUB 2550:GOSUB 2470:J=0:SWAP PS(I,0),PS(I,1)
830 W=PS(I,0):GOSUB 2540:GOSUB 2480:GOSUB 2650:RETURN
840 IF E=0 THEN 730
850 IF E<2 THEN 980
860 LE=E+1:FOR J=0 TO E:LT(J+1)=J:NEXT J
870 FOR J=1 TO LE:W=PC(I,LT(J)):GOSUB 2510:WN(J)=N:WS(J)=S:NEXT J
880 IF WN(LE)=14 THEN LEF=LE-1 ELSE LEF=LE
890 FOR J=1 TO LEF:IF WN(J)=WN(1) THEN NEXT J:GO TO 940
900 FOR J=1 TO LEF:IF WS(J)<>WS(1) THEN 980 ELSE NEXT J
910 IF WN(LE)=14 THEN FOR J=2 TO LEF:IF WN(J)-WN(J-1)=2 THEN WN(LE)=WN(J)-1 ELSE
 NEXT J:GO TO 930
920 LEF=LE:FOR J=1 TO LEF-1:IF WN(J)>WN(J+1) THEN SWAP WN(J),WN(J+1):NEXT J ELSE
 NEXT J
930 FOR J=2 TO LEF:IF WN(J)-WN(J-1)<>1 THEN 980 ELSE NEXT J
940 FOR J=1 TO LE:X=(LT(J))*53:Y=(I+1)*40:LTJ=LT(J):GOSUB 2550:XD(I)=XD(I)+1:SC(
I,XD(I))=PC(I,LT(J)):PC(I,LTJ)=60:NEXT J:FOR J=0 TO E:GOSUB 2470:NEXT J
950 GOSUB 2210:E(I)=E(I)-LE:E=E(I)
960 FOR J=0 TO XD(I):W=SC(I,J):GOSUB 2480:X=583-53*J:Y=(I+1)*40:GOSUB 2650:NEXT
J
970 GOSUB 2570:GO TO 1770
980 GOSUB 2570:LOCATE0,22:PRINT'Cannot finish, try again !':FOR J=0 TO 1000:NEXT
J:GOSUB 2570:GO TO 420
990 RANDOMIZE(TIME/10):J=INT(RND(1)*2+1):ON J GOTO 1000,1020
1000 IF A<2 THEN 1020 ELSE IF A>3 THEN A=2
1010 LE=A+1:FOR J=0 TO A:LT(J+1)=TS(J):NEXT J:GO TO 1040
1020 IF B<2 THEN 1000 ELSE IF B>4 THEN B=2
1030 LE=B+1:FOR J=0 TO B:LT(J+1)=TN(J):NEXT J:GO TO 1040
1040 FOR J=1 TO LE:LTJ=LT(J):XD(I)=XD(I)+1:SC(I,XD(I))=PC(I,LT(J)):PC(I,LTJ)=60:
NEXT J:E(I)=E(I)-LE:E=E(I)
1050 GOSUB 2530:GOSUB 2210
1060 FOR J=0 TO XD(I):W=SC(I,J):GOSUB 2480:X=583-53*J:Y=(I+1)*40:GOSUB 2650:NEXT
 J
```

```
1070 IF E<0 THEN 1480 ELSE RETURN
1080 IF XD(K)<0 THEN RETURN
1090 FOR J=0 TO XD(K):SCN(K,J)=0:SCS(K,J)=0
1100 W=SC(K,J):GOSUB 2510:SCN(K,J)=N:SCS(K,J)=S:NEXT J:LC=0
1110 LJ=-1:FOR L=LC TO LC+2:IF SCN(K,L)=14 THEN LJ=L
1120 NEXT L:IF LJ<>-1 THEN 1240
1130 IF SCN(K,LC)=SCN(K,LC+1) AND SCN(K,LC)=SCN(K,LC+2) THEN 1160
1140 IF SCN(K,LC)-SCN(K,LC+1)=1 AND SCN(K,LC+1)-SCN(K,LC+2)=1 THEN 1190
1150 LC=LC+3:IF LC>XD(K) THEN RETURN ELSE 1110
1160 IF SCN(K,LC)=SCN(K,LC+3) AND (XD(K)+1)MOD3<>0 THEN LN=4 ELSE LN=3
1170 IF SWN=SCN(K,LC) THEN LD=LC:GO TO 1420
1180 LC=LC+LN:IF LC>XD(K) THEN RETURN ELSE 1110
1190 IF SCN(K,LC+2)-SCN(K,LC+3)=1 AND (XD(K)+1)MOD3<>0 THEN LN=4 ELSE LN=3
1200 IF SWS<>SCS(K,LC) THEN 1230
1210 IF SWN-SCN(K,LC)=1 THEN LD=LC:GO TO 1420
1220 IF SCN(K,LC+LN-1)-SWN=1 THEN LD=LC+LN:GO TO 1420
1230 LC=LC+LN:IF LC>XD(K) THEN RETURN ELSE 1110
1240 IF LJ<>LC THEN 1250 ELSE IF SCN(K,LC+1)=SCN(K,LC+2) THEN 1270 ELSE 1310
1250 IF LJ<>LC+1 THEN 1260 ELSE IF SCN(K,LC)=SCN(K,LC+2) THEN 1270 ELSE 1310
1260 IF SCN(K,LC)=SCN(K,LC+1) THEN 1270 ELSE 1310
1270 IF LJ=LC THEN LE=LC+1 ELSE LE=LC
1280 IF SCN(K,LE)=SCN(K,LC+3) AND (XD(K)+1)MOD3<>0 THEN LN=4 ELSE LN=3
1290 IF SWN=SCN(K,LE) THEN LD=LC:GO TO 1420
1300 LC=LC+LN:IF LC>XD(K) THEN RETURN ELSE 1110
1310 IF LJ<>LC THEN 1330 ELSE IF SCN(K,LC+1)-SCN(K,LC+2)=1 THEN 1350
1320 IF SCN(K,LC+1)-SCN(K,LC+2)=2 THEN 1350 ELSE 1410
1330 IF LJ<>LC+1 THEN 1340 ELSE IF SCN(K,LC)-SCN(K,LC+2)=2 THEN 1350 ELSE 1410
1340 IF SCN(K,LC)-SCN(K,LC+1)=1 THEN 1350 ELSE 1410
1350 IF LJ=LC+2 THEN LF=2:LE=LC+1 ELSE LF=1:LE=LC+2
1360 IF SCN(K,LC)-SCN(K,LE+3)=LF AND SCS(K,LE)=SCS(K,LC+3) THEN LN=4 ELSE LN=3
1370 LE=LC+LN-1:LF=1:IF LJ=LE THEN LF=2:LE=LC+LN-2
1380 IF SWN-SCN(K,LE)=-LF AND SWS=SCS(K,LE) THEN LD=LE+1:GOTO 1420
1390 LE=LC:LF=1:IF LJ=LE THEN LF=2:LE=LC+1
1400 IF SWN-SCN(K,LE)=LF AND SWS=SCS(K,LE) THEN LD=LE:GOTO 1420
1410 LC=LC+LN:IF LC>XD(K) THEN RETURN ELSE 1110
1420 EN=1:IF TES=1 THEN RETURN
1430 FOR L=XD(K) TO LD STEP -1:SC(K,L+1)=SC(K,L):NEXT L
1440 SC(K,LC)=PC(I,SW):XD(K)=XD(K)+1
1450 X=SW*53:Y=(I+1)*40:GOSUB 2550:PC(I,SW)=60:E(I)=E(I)-1:E=E(I)
1460 FOR J=0 TO XD(K):W=SC(K,J):GOSUB 2480:X=583-53*J:Y=(K+1)*40:GOSUB 2650:NEXT
 J
1470 GOSUB 2570:J=SW:GOSUB 2540:GOSUB 2550:IF E<0 THEN 1480 ELSE RETURN
1480 LINE (0,(I+1)*40)-(50,(I+1)*40+37),PSET,0,BF
1490 GO TO 1770
1500 FOR SW=0 TO E:W=PC(I,SW):GOSUB 2510:SWN=N:SWS=S:FOR K=0 TO 2:GOSUB 1080
1510 IF EN<>1 THEN NEXT K,SW:GO TO 1540
1520 IF SW=0 THEN SWAP PC(I,0),PC(I,1):J=1:GOSUB 2540:GOSUB 2550
1530 J=0:GOSUB 2540:GOSUB 2580:RETURN
1540 TES=1:W=TC(DT-1):GOSUB 2510:SWN=N:SWS=S:FOR K=0 TO 2:GOSUB 1080
1550 IF EN<>1 THEN NEXT K:TES=0:RETURN
1560 SWAP TC(DT-1),PC(I,0):J=E(I):GOSUB 2540:GOSUB 2580:W=TC(DT-1):GOSUB 2520:RE
TURN
1570 TW=TC(22):DT=22:FOR M=0 TO 5:RANDOMIZE(TIME/3):MD=INT(RND(1)*20):RANDOMIZE(
TIME/5):MU=INT(RND(1)*20):IF MD>MU THEN SWAP MD,MU
1580 FOR MI=MU TO DT-1:SWAP TC(MI),TC(MD+MI-MU):NEXT MI,M
1590 FOR MI=1 TO DT:DC(MI)=TC(MI-1):NEXT MI:DT=1:DN=1:TC(0)=TW:RETURN
1600 COLOR4:LOCATE 55,22:PRINT "[";:COLOR2:PRINT "RETURN";:COLOR4:PRINT "]";:COL
OR6:PRINT " Show Cards"
1620 COLOR4:LOCATE 55,24:PRINT "[";:COLOR2:PRINT " INS. ";:COLOR4:PRINT "]";:COL
OR6:PRINT " Insert Card";
1630 COLOR4:LOCATE 60,0:PRINT "1 POINT = ¥";:COLOR2:PRINT POI;
1640 COLOR7:LOCATE 59,1:PRINT "PLAYER 1 = ¥";:COLOR2:PRINT USING "#####";MNE(0);
1650 COLOR7:LOCATE 59,2:PRINT "PLAYER 2 = ¥";:COLOR2:PRINT USING "#####";MNE(1);
1660 COLOR7:LOCATE 59,3:PRINT "PLAYER 3 = ¥";:COLOR2:PRINT USING "#####";MNE(2);
:RETURN
1670 COLOR2:LOCATE5,8:PRINT "POINTS :":X=16:Y=8:COLOR5:ME=15:GOSUB 1760
1680 FOR ME=2 TO 7:X=X+7:GOSUB 1760:NEXT ME
1690 COLOR2:LOCATE5,13:PRINT "POINTS :":X=9:Y=13:COLOR5
1700 FOR ME=8 TO 9:X=X+7:GOSUB 1760:NEXT ME
1710 FOR MIE=0 TO 3:X=X+7:ME=10:GOSUB 1760:NEXT MIE
1720 X=X+6:ME=20:GOSUB 1760
1730 IF PON=0 THEN 1750 ELSE COLOR6:LOCATE16,10:INPUT "Rebet your money (Y/N) ";
PO$
1740 IF PO$="Y" THEN 1750 ELSE RETURN
1750 COLOR6:LOCATE16,10:INPUT "Bet how much for 1 point ";POI:IF POI>1000 THEN G
OTO 1750
1760 LOCATE X,Y:PRINT ME;:RETURN
1770 GOSUB 2570:FOR J=0 TO 7:BEEP1:LOCATE 0,22:PRINT "GAME OVER !";:FOR J1=0 TO
20:NEXTJ1:BEEP0:LOCATE0,22:PRINTSPC(11);:FOR J1=0 TO 10:NEXT J1,J:CLS:PON=1
1780 COLOR4:LOCATE30,0:PRINT " 1 POINT =¥";POI
1790 COLOR3:LOCATE30,1:PRINT "        A = 15"
1800 COLOR5:LOCATE30,2:PRINT "      1-9 = 1-9"
1810 COLOR6:LOCATE30,3:PRINT "10,J,Q,K = 10"
1820 COLOR2:LOCATE30,4:PRINT "   JOCKER = 20"
1830 X=0:Y=50:FOR K=0 TO 2:IF K=I THEN NEXT K:GO TO 1910
1840 P=0:FOR J=0 TO E(K):W=PC(K,J):GOSUB 2480:GOSUB 2650
1850 IF N=14 THEN P=P+20:GO TO 1890
1860 IF N=1 THEN P=P+15:GO TO 1890
1870 IF N>=10 AND N<=13 THEN P=P+10:GO TO 1890
1880 P=P+N
1890 X=X+53:NEXT J:GOSUB 1980
1900 Y=Y+70:X=0:NEXT K:EN=0
1910 LOCATE 0,23:INPUT "Continue to play (Y/N) ";P$
```

```
1920 IF P$<>"Y" THEN 1950
1930 FOR K=0 TO 2:IF MNE(K)<0 THEN 1950
1940 NEXT K:CLS:GO TO 40
1950 MN$="PLAYER":CLS:FOR K=0 TO 2:IF MNE(K)<0 THEN MNE$=MN$+STR$(K+1)+" IS BANK
RUPT":GOSUB 1970:NEXT K:END
1960 MNE$=MN$+STR$(K+1)+"=¥"+STR$(MNE(K)):GOSUB 1970:NEXT K:END
1970 SYMBOL (50,K*50+30),MNE$,3,4,K+4:RETURN
1980 COLOR7:LOCATE0,(Y+40)¥8+1:PRINT "Player";K+1;"has to pay ";:COLOR2:PRINT "¥
";POI*P;:MNE(K)=MNE(K)-POI*P:MNE(I)=MNE(I)+POI*P
1990 COLOR7:PRINT ",owns ¥";MNE(K):RETURN
2000 A=0:B=0:IF E>6 THEN 2120 ELSE IF PC(I,E)<>52 THEN 2120
2010 GOSUB 2220:FOR A=0 TO 9:TN(A)=0:TS(A)=0:NEXT A:A=-1:B=-1
2020 FOR J=E-1 TO 0 STEP-1:IF CS(I,J)=1 THEN GOSUB 2080
2030 NEXT J:FOR J=E-1 TO 0 STEP-1:IF CN(I,J)=>2 THEN GOSUB 2090
2040 NEXT J:IF A>1 THEN A=1
2050 IF B>2 THEN B=1
2060 IF A<0 THEN 2070 ELSE A=A+1:TS(A)=E:RETURN
2070 IF B<0 THEN RETURN ELSE B=B+1:TN(B)=E:RETURN
2080 A=A+1:TS(A)=J:RETURN
2090 M=CN(I,J)-1:WA=J:GOSUB 2110
2100 FOR K=J-1 TO 0 STEP -1:IF CN(I,K)=M AND PS(I,K)=PS(I,J) THEN M=M-1:IF M<0 T
HEN RETURN ELSE WA=K:GOSUB 2110:NEXT K:RETURN ELSE NEXT K:RETURN
2110 B=B+1:TN(B)=WA:RETURN
2120 GOSUB 2220:FOR A=0 TO 9:TN(A)=0:TS(A)=0:NEXT A:A=-1:B=-1
2130 FOR J=E TO 0 STEP-1:IF CS(I,J)>=2 THEN GOSUB 2160
2140 NEXT J:FOR J=E TO 0 STEP-1:IF CN(I,J)>2 AND CN(I,J)<5 THEN GOSUB 2170
2150 NEXT J:RETURN
2160 A=A+1:TS(A)=J:RETURN
2170 IF B<0 THEN 2180 ELSE FOR L=0 TO B:IF TN(L)=J THEN RETURN ELSE NEXT L
2180 M=CN(I,J)-1:WA=J:GOSUB 2200
2190 FOR K=J-1 TO 0 STEP -1:IF CN(I,K)=M AND PS(I,K)=PS(I,J) THEN M=M-1:IF M<0 T
HEN RETURN ELSE WA=K:GOSUB 2200:NEXT K:RETURN ELSE NEXT K:RETURN
2200 B=B+1:TN(B)=WA:RETURN
2210 LINE ((E+1)*53,(I+1)*40)-(689-XD(I)*53,(I+1)*40+37),PSET,0,BF:RETURN
2220 FOR J=0 TO E:CN(I,J)=0:CS(I,J)=0:NEXT J
2230 KS=0:FOR J=0 TO E:W=PC(I,J):GOSUB 2510:PN(I,J)=N:PS(I,J)=S:NEXT J
2240 FOR J=0 TO E-1:CN1=0:IF KS<>0 THEN KS=KS-1:GO TO 2310
2250 IF PS(I,J)>2 THEN 2310
2260 KJ=J+3:IF J=E-2 OR J=E-1 THEN KJ=E
2270 KS=0:FOR K=J+1 TO KJ:IF PN(I,J)=PN(I,K) THEN KS=KS+1
2280 NEXT K:IF KS=3 THEN FOR L=J TO J+KS:CS(I,L)=3:NEXT L
2290 IF KS=0 THEN 2310 ELSE IF KS<3 THEN FOR L=J TO J+KS:CS(I,L)=2:NEXT L
2300 IF KS<2 THEN FOR L=J TO J+KS:CS(I,L)=1:NEXT L
2310 FOR K=J+1 TO E:IF PS(I,K)<>PS(I,J) THEN NEXT K,J:GO TO 2370
2320 U=PN(I,K)-PN(I,J)
2330 IF PN(I,J)*(U-1)=0 THEN IF CN(I,K)=0 THEN CN(I,J)=1:CN(I,K)=2:NEXT K,J:GO T
O 2370
2340 IF PN(I,J)*(U-2)=0 THEN IF CN(I,J)=1 THEN CN(I,K)=3:CN1=1:NEXT K,J:GO TO 23
70
2350 IF PN(I,J)*(U-3)=0 THEN IF CN(I,J)=1 AND CN1=1 THEN CN(I,K)=4:CN1=0:NEXT K,
J:GO TO 2370
2360 NEXT K:NEXT J
2370 IF PC(I,E)=52 THEN CN(I,E)=5
2380 RETURN
2390 IJ=0:FOR I=0 TO 2:FOR J=0 TO 9:PC(I,J)=CARD(IJ):IJ=IJ+1:NEXT J,I
2400 FOR I=0 TO 22:DC(I)=CARD(I+30):NEXT
2410 FOR I=0 TO 2:GOSUB 2530:NEXT I
2420 FOR J=0 TO 9:FOR I=0 TO 2:X=J*53:Y=(I+1)*40
2430 IF I=2 THEN W=PC(I,J):GOSUB 2480:GOSUB 2650 ELSE GOSUB 2580
2440 NEXT I,J:FOR J=0 TO 9:GOSUB 2470:NEXT
2450 COLOR4:LOCATE7,2:PRINT"THROWN CARDS":W=DC(0):TC(0)=DC(0):DT=1:DN=1:GOSUB 25
20
2460 GOSUB 2560:LOCATE 40,2:PRINT"DRAW CARDS":FOR I=0 TO 2:XD(I)=-1:E(I)=9:NEXT
I:RETURN
2470 SYMBOL (J*53+14,163),STR$(J),1,1,4,,XOR:RETURN
2480 S$=CHR$(232+W MOD4):N=W¥4+1
2490 IF W MOD4=0 OR W MOD4=3 THEN C=0 ELSE C=2
2500 N$=RIGHT$(STR$(N),1):RETURN
2510 S=W MOD4:N=W¥4+1:RETURN
2520 X=159:Y=0:GOSUB 2480:GOSUB 2650:RETURN
2530 FOR K=0 TO 8:FOR J=0 TO 8:IF PC(I,J)>PC(I,J+1) THEN SWAP PC(I,J),PC(I,J+1):
NEXT J,K:RETURN ELSE NEXT J,K:RETURN
2540 X=J*53:Y=(I+1)*40:RETURN
2550 FOR Z=0 TO 20:BEEP1:NEXTZ:BEEP0:LINE (X,Y)-(X+50,Y+37),PSET,0,BF:RETURN
2560 X=265:Y=0:GOSUB 2580:RETURN
2570 LOCATE 0,22:PRINT SPC(50):RETURN
2580 FOR Z=0 TO 20:BEEP1:NEXTZ:BEEP0:LINE (X,Y)-(X+50,Y+37),PSET,7,BF
2590 LINE (X+6,Y+3)-(X+44,Y+34),PSET,0,BF
2600 FOR Y1=Y+3 TO Y+27 STEP 8:FOR X1=X+6 TO X+41 STEP 8
2610 SYMBOL (X1,Y1),CHR$(254),1,1,4:NEXT X1,Y1
2620 FOR Y1=Y+11 TO Y+31 STEP 10:FOR X1=X+12 TO X+32 STEP 10
2630 SYMBOL (X1,Y1),"#",1,1,0,1:NEXT X1,Y1
2640 RETURN
2650 FOR Z=0 TO 20:BEEP1:NEXTZ:BEEP0:LINE (X,Y)-(X+50,Y+37),PSET,7,BF
2660 N$=RIGHT$(STR$(N),1)
2670 ON N GOTO 2690,2730,2730,2800,2800,2890,2890,2890,3020,3020,3210,3280,3350,
3420
2680 RETURN
2690 SYMBOL (X+2,Y+1),"A",1,1,C
2700 SYMBOL (X+16,Y+16),S$,2,1,C
2710 SYMBOL (X+48,Y+36),"A",1,1,C,2
2720 RETURN
```

```
2730 SYMBOL (X+2,Y+1),N$,1,1,C
2740 SYMBOL (X+21,Y+5),S$,1,1,C
2750 IF N=2 THEN 2770
2760 SYMBOL (X+21,Y+16),S$,1,1,C
2770 SYMBOL (X+29,Y+33),S$,1,1,C,2
2780 SYMBOL (X+48,Y+36),N$,1,1,C,2
2790 RETURN
2800 SYMBOL (X+2,Y+1),N$,1,1,C
2810 SYMBOL (X+12,Y+5),S$,1,1,C
2820 SYMBOL (X+30,Y+5),S$,1,1,C
2830 IF N=4 THEN 2850
2840 SYMBOL (X+21,Y+16),S$,1,1,C
2850 SYMBOL (X+20,Y+32),S$,1,1,C,2
2860 SYMBOL (X+38,Y+32),S$,1,1,C,2
2870 SYMBOL (X+48,Y+36),N$,1,1,C,2
2880 RETURN
2890 SYMBOL (X+2,Y+1),N$,1,1,C
2900 SYMBOL (X+12,Y+5),S$,1,1,C
2910 SYMBOL (X+30,Y+5),S$,1,1,C
2920 SYMBOL (X+12,Y+16),S$,1,1,C
2930 SYMBOL (X+30,Y+16),S$,1,1,C
2940 IF N=6 THEN 2980
2950 SYMBOL (X+21,Y+10),S$,1,1,C
2960 IF N=7 THEN 2980
2970 SYMBOL (X+29,Y+28),S$,1,1,C,2
2980 SYMBOL (X+20,Y+33),S$,1,1,C,2
2990 SYMBOL (X+38,Y+33),S$,1,1,C,2
3000 SYMBOL (X+48,Y+36),N$,1,1,C,2
3010 RETURN
3020 IF N=10 THEN 3040
3030 SYMBOL (X+2,Y+1),N$,1,1,C:GO TO 3060
3040 SYMBOL (X,Y+1),"1",1,1,C
3050 SYMBOL (X+5,Y+1),"0",1,1,C
3060 SYMBOL (X+12,Y+3),S$,1,1,C
3070 SYMBOL (X+30,Y+3),S$,1,1,C
3080 SYMBOL (X+12,Y+11),S$,1,1,C
3090 SYMBOL (X+30,Y+11),S$,1,1,C
3100 SYMBOL (X+21,Y+7),S$,1,1,C
3110 SYMBOL (X+20,Y+26),S$,1,1,C,2
3120 SYMBOL (X+38,Y+26),S$,1,1,C,2
3130 SYMBOL (X+20,Y+34),S$,1,1,C,2
3140 SYMBOL (X+38,Y+34),S$,1,1,C,2
3150 IF N=10 THEN 3170
3160 SYMBOL (X+48,Y+36),N$,1,1,C,2:RETURN
3170 SYMBOL (X+29,Y+28),S$,1,1,C,2
3180 SYMBOL (X+50,Y+36),"1",1,1,C,2
3190 SYMBOL (X+45,Y+36),"0",1,1,C,2
3200 RETURN
3210 SYMBOL (X+2,Y+1),"J",1,1,C
3220 SYMBOL (X+2,Y+9),S$,1,1,C
3230 SYMBOL (X+48,Y+28),S$,1,1,C,2
3240 SYMBOL (X+48,Y+36),"J",1,1,C,2
3250 LINE (X+12,Y+5)-(X+39,Y+32),PSET,0,B
3260 PUT@ (X+14,Y+7)-(X+37,Y+30),AJ,PSET,C
3270 RETURN
3280 SYMBOL (X+2,Y+1),"Q",1,1,C
3290 SYMBOL (X+2,Y+9),S$,1,1,C
3300 SYMBOL (X+48,Y+28),S$,1,1,C,2
3310 SYMBOL (X+48,Y+36),"Q",1,1,C,2
3320 LINE (X+12,Y+5)-(X+39,Y+32),PSET,0,B
3330 PUT@ (X+14,Y+7)-(X+37,Y+30),AQ,PSET,C
3340 RETURN
3350 SYMBOL (X+2,Y+1),"K",1,1,C
3360 SYMBOL (X+2,Y+9),S$,1,1,C
3370 SYMBOL (X+48,Y+28),S$,1,1,C,2
3380 SYMBOL (X+48,Y+36),"K",1,1,C,2
3390 LINE (X+12,Y+5)-(X+39,Y+32),PSET,0,B
3400 PUT@ (X+14,Y+7)-(X+37,Y+30),AK,PSET,C
3410 RETURN
3420 SYMBOL (X+2,Y+1),"X",1,1,3
3430 SYMBOL (X+2,Y+1),"O",1,1,3
3440 SYMBOL (X+48,Y+36),"O",1,1,3,2
3450 SYMBOL (X+48,Y+36),"X",1,1,3,2
3460 LINE (X+12,Y+5)-(X+39,Y+32),PSET,0,B
3470 PUT@ (X+14,Y+7)-(X+37,Y+30),AJO,PSET,3
3480 RETURN
3490 DATA 8514,-31327,17034,22398,-3000,8,16383,-2006,-16368,10945
3500 DATA -12203,-30960,21899,-5975,-30872,-22134,-19287,-32504,20864,1105
3510 DATA -32460,20866,27729,-31780,30578,-1810,-8144,21824,8277,30752
3520 DATA -10825,-8083,-10816,987,24577,32312
3530 DATA 10923,-7125,-20612,21929,-9131,-24196,-21568,31915,24,22016
3540 DATA 6236,14808,-20992,6313,984,22558,-10152,-7444,-22526,13364
3550 DATA 308,13326,27222,106,22047,22231,7772,-10490,-25937,-6505
3560 DATA 22513,-10537,-307,-20518,-19002,21866
3570 DATA -31728,9047,-4115,12271,-9429,-2,3072,1567,-2,5120
3580 DATA 11029,-14357,5152,11031,-11349,5447,-21680,16469,17480,11020
3590 DATA 6427,3160,5428,8235,13436,11060,4269,21560,-21675,-24149
3600 DATA 21930,-10917,-21835,-12321,-4729,-14795
3610 DATA 144,0,-22016,362,1,27152,987,-26622,-19932,484
3620 DATA -31741,-21852,513,-23550,-7852,1537,-21997,-32344,-957,-12284
3630 DATA -7772,2314,8221,-864,2730,-24563,21776,31381,-1981,-1172
3640 DATA 10020,27731,22070,26026,22557,-19880
```

SHINKO

時代をリードする富士通のOA機器ショップ

高級ホビーからビジネスまでの多才パソコン

FM-8

¥218,000（本体MB25020）

FM-7

新しい感性を伝えてくれる、FM-7。
マニアのために、新登場。

¥126,000（本体MB25010簡易言語ソフト付）

ビジネス用途を大きく拡げる高級パソコン

FM-11

■FM-11EX **¥398,000**（本体MB25050 CP／M-86 簡易言語ソフト付）
■FM-11AD **¥338,000**（本体MB25040 簡易言語ソフト付）
■FM-11ST **¥268,000**（本体MB25030）

※テレビは別売です。

ハードからソフトウェアまで、システム開発設計。

■営業案内

●マイクロコンピュータ・パーソナルコンピュータ・半導体製品及び電子部品のご相談販売
マイコンチップ・TTL・リニアIC・コネクタ等電子部品の販売及びアセンブリについてご相談に応じます。
●マイクロコンピュータ応用システムの開発ご相談
マイクロコンピュータを使用した制御機器の設計及びソフトウエア開発等のご相談に応じます。
●パーソナルコンピュータ応用システムの開発、ご相談
パーソナルコンピュータを使用したシステムの開発とご相談に応じます。

■取扱い製品

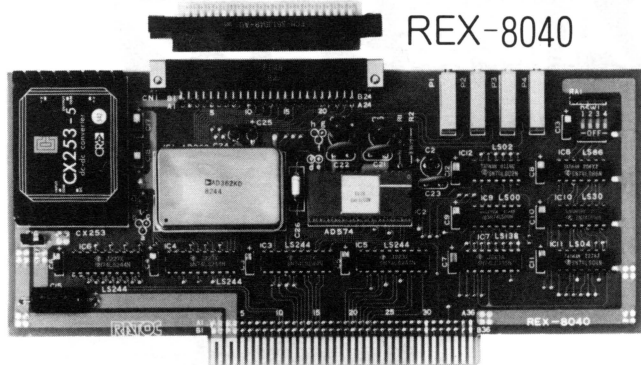
●マイクロコンピュータ
●メモリIC
●セミカスタムLSI
●ロジックIC
●リニアIC
●コネクタ
●リレー
●電子部品、他

■販売代理店募集 詳しくは新光コンピュータサービスへお問い合せ下さい。

新光電気工業株式会社
本社／〒380 長野市栗田711
TEL0262－26－1145㈹

新光コンピュータサービス
事務所／〒380 長野市南石堂町1236 TEL0262－28－2443
営業時間：午前8：40～午後5：30
（日曜、祝日、第二、第四土曜日は定休日）

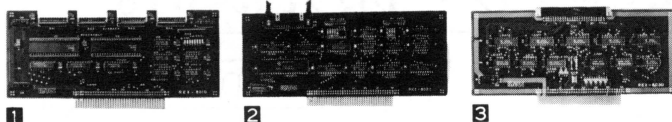
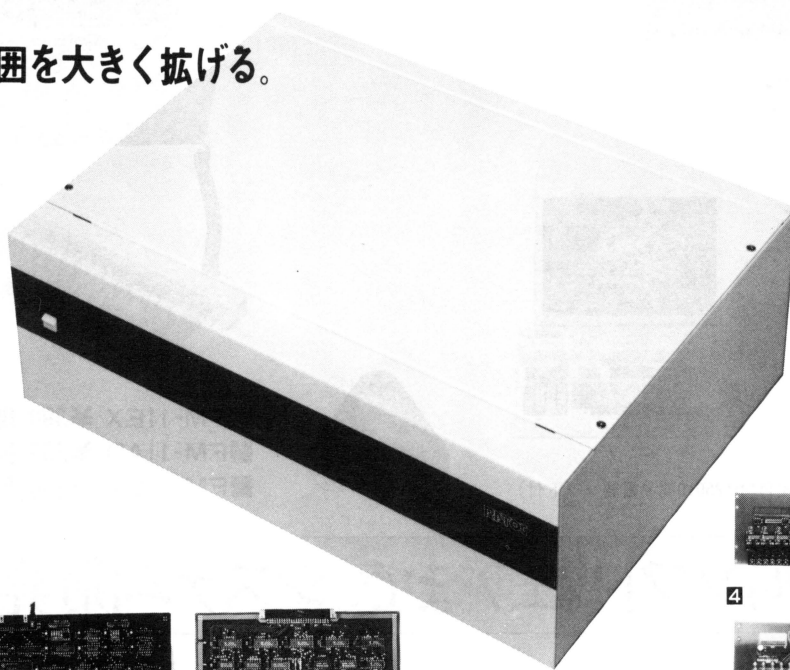
ラボラトリ・オートメーション

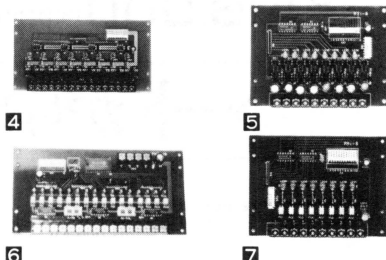
LA by FM-7 & 8

A/D変換カード REX-8040

好・評・発・売・中 ￥198,000

- ■入力チャンネル 8ch(差動)、16ch(シングルエンド)
- ■入力レンジ……………………0～10V、−10V～＋10V
- ■解像度……………………………………………12bit
- ■変換時間…………………………………………35vs
- ■アクイジション……………………………………18vs

FM-7、8の応用範囲を大きく拡げる。

▨ I/O拡張ユニット
REX-8 ￥83,000

REX-8オプションカード

- 1 PIAカード REX-8010……………………￥ 23,500
- 2 GPIBカード REX-8020…………………￥ 60,000
- 3 電子天秤 I/Fカード REX-8030…………￥ 35,000
- ■D/A変換カード REX-8050………………￥138,000

PiAカード用アイソレーション入出力ユニット

- 4 リードリレー出力ユニット(8 bit)……………￥17,500
- 5 リードリレー入力ユニット(8 bit)……………￥16,000
- 6 フォトアイソレート出力ユニット(16chデコード)￥17,500
- 7 フォトアイソレート入力ユニット(8 bit)………￥14,000

RATOC

ラトック システム エンジニアリング株式会社

プロダクツ事業部：大阪市西区江之子島1−6−2 奥内8号ビル
TEL. 06−445−1657(代)

● 販売代理店

関西：
- ●コムスポット共立 TEL 06 (644) 4666
- ●㈱エルム・システム営業部 TEL 06 (204) 0611
- ●ダイワボウ情報システム㈱ TEL 06 (281) 1161
- ●アドコム電子㈱ TEL 075 (939) 5231
- ●マイクロコンピュータシステムズ TEL 06 (358) 3045
- ●関東電子㈱／京都営業所 TEL 075 (343) 0995
- ●上新電機㈱ J & P TEL 06 (644) 1413

関東：
- ●㈱アスターインターナショナル TEL 03 (257) 0128
- ●コスモス 秋葉原 TEL 03 (257) 0512
- ●マイクロコミュニケーションズ TEL 03 (350) 1174
- ●MKショップ 和知電子 TEL 03 (255) 7431
- ●亜土マイコンショップ TEL 03 (255) 9515
- ●サフジ電子部品販売㈱ TEL 03 (255) 7581

東北：
- ●東北電子産業㈱ TEL 0222 (66) 1681

より高度化する情報化社会に向けて

ソフトウェアからハードウェアまで

使いやすさとトータル・パフォーマンスを実現しました。

あなたの会社に合わせた財務会計システム

〈監修〉 公認会計士 吉村 成弘　　〈開発・販売〉 ピーシーエー株式会社，都築電産株式会社

〈特 長〉

○会計専門家数名のノウハウを結集したシステムです。
○あなたの会社に合わせた科目登録や補助簿登録ができます。
○主要帳票はオフコン並の専用用紙が用意されておりますので、銀行や税務署等にそのまま提出できます。
○画面は漢字の大文字で表示され、非常に見やすくなっています。
○日常良く使われる主要48科目は、ワンタッチ入力方式になっています。
○入出金データーの入力は自動仕訳方式にする等、使いやすく、正確で、しかも主要部分は機械語処理により、トータル・パフォーマンスを追求しています。
○システムAからCへのレベルアップが可能です。

〈システムの概要〉

項目	システム A	B	C	項目	システム A	B	C	項目	システム A	B	C
I 基本設計事項				勘定科目	○	○	○	元　帳（仮締、本締）	○	○	○
会計処理月等の登録	○	○	○	金　額	○	○	○	補助元帳（仮締、本締）		○	○
伝票No.の入力有無	○	○	○	検索データーの訂正・取消し	○	○	○	補助科目残高一覧表		○	○
主勘定科目数 240	○	○	○	**IV 画面表示**				**合計残高試算表**			
補助科目数 240		○	○	漢字による操作指示	○	○	○	B/S（仮締、本締）	○	○	○
部門設定 9			○	検索データー	○	○	○	P/L（仮締、本締）	○	○	○
摘要コード 100	○	○	○	元　帳	○	○	○	資金繰実績表			○
各項目の登録、訂正機能	○	○	○	合計残高試算表	○	○	○	経営分析表			○
月末、期末自動更新	○	○	○	**V 作成帳表**				部門別損益計算書			○
II データー入力方法				元帳の科目指定プリント	○	○	○	部門別P/L一覧表			○
入出金データーの自動仕訳	○	○	○	期間指定プリント				月次残高推移表			○
単一振替伝票	○	○	○	仕訳日記帳	○	○	○	P/L対比棒グラフ			○
複合振替伝票	○	○	○	日計表		○	○	手形管理表		○	○
主要48科目ワンタッチ入力	○	○	○	元　帳	○	○	○	取引先別総合収支明細表			○
常時1件前の入力データー表示	○	○	○	補助元帳		○	○	**決算報告書**			
データーの訂正、取消し	○	○	○	合計残高試算表	○	○	○	貸借対照表	○	○	○
III データーの検索				登録データーチェックリスト	○	○	○	損益計算書	○	○	○
（仕訳データーのチェック用）				入力データーチェックリスト	○	○	○	販売管理費明細書	○	○	○
伝票番号	○	○	○	仕訳日記帳	○	○	○	原価計算報告書	○	○	○
日　付	○	○	○	日計表		○	○	利益処分計算書	○	○	○

〈システムの価格〉 A（元帳、試算表、決算書）￥98,000
B（A＋補助簿、手形管理）￥148,000
C（B＋部門管理、その他分析）￥198,000

販売代理店募集

※詳しくは本社第一販売部へお問い合わせください。

●お問い合わせは 富士通特約店 TDC 都築電産株式会社 〒105 東京都港区西新橋3丁目11番6号千政ビル TEL（03）433-2171（代）

関西支店 大阪市淀川区宮原2丁目11番9号 （06）393-0881
名古屋営業所 名古屋市中区錦2丁目9番1号 （052）201-4768
ショールーム （東京）（03）433-2171
京都営業所 京都市南区東9条南河辺町39番 （075）671-0345
東北営業所 郡山市清水台2丁目11番14号 （0249）23-6350
ショールーム （大阪）（06）393-0881（代）

FM-8&FM-7 Disk Utility
HOSPITAL

健康診断
ディスケット上に機械的なキズ(Physical Error)がないかどうか調べます。全トラックについて読みだしテストを行ない、必要に応じてFormat・システムコピー等をする事が出来ます。

胃腸科
F-Basic用ディスクのFAT/Direetryを解読します。FAT/Direetryの指定のほかに"FILES"の指定をすると、Killされたファイルも含めて、ファイル名・クラスタ使用状況の一覧表が示されます。

X線診断
FM-8仕様に限らず、両倍(40T)・片倍(35T)ディスケットの任意のセクタがスクリーン上に示されます。他社仕様のディスクを読んで理解するためには、その社の呼称法、セクターの読み出し順序の法則を、F-Basicでの法則に換算しながら進めて下さい。

整形外科
上と同じく、任意のセクタをバイト単位でスクリーン上で修正できます。このルーチンも無意味なトラックを指定しない限り他社仕様のディスクも対象となります。間違って入力しても(CR)しない限り何度でも修正出来ます。

脳神経外科
プログラムの収納の構造が調べられ、ツブれたプログラムの修復・他社仕様のディスケットの移植の準備に使われます。各クラスタの最初のセクタを表示します。また、このセクタの前半分と直前のクラスタの最後の半セクタを表示することや、直前のセクタのつながりを見たり、プリンタがあればCopyを取ったり、新しいクラスタの指示も行なう事が出来ます。

価格 15,000円
〒300

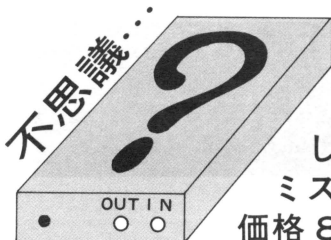
産婦人科
F-Basic用のFormat・システムコピー・初期化とVolcopy等を実行します。Format・Volcopyは、他機種、他言語にも使えます。いずれの場合も、#0上のプログラムディスケットが読み込まれ実行が始まります。#0に入っているプログラムディスケットを、出来るだけ出し入れしないようにしていますのでFujitsuから提供されるものよりは作業が簡略化されています。

人工受精
片倍(35T)ディスケットの任意の面の全Copyに使います。表面(例えばPC仕様の片面)、裏面(例えばIf-800)が指定できます。片倍(35T)のディスケットは普通反対面が、Format していないので、通常のVolcopyが出来ません。この様な場合、表あるいは裏を指定して、その面を35トラックだけVolcopyするルーチンです。尚、PASOPIAのように両面で35Tの場合は、このルーチンを表裏2度走らせば良いのです。このプログラムでも、産婦人科でもVolcopyは、Killされたファイルも含めて、内容を判断する事なく、完全Copyします。

心臓移植
PC-8801(NEC)の両倍仕様の、任意の一つのプログラム・ファイルをFM-8仕様にCopyします。それに先立ってPC-8801仕様のディスケットの構造を、調べる事が出来ます。このルーチンのうちファイルコピープログラムそのものは、オリジナルがAscii Saveされていなくても動きますが、目的が移植にあるわけですから、オリジナルのAscii Saveは必項です。このプログラムではドライブ番号、トラック番号等をいきなりFM-形で書かないで、一度PC(両面)形で書いて改めて換算するようにしています。また、片面倍密度仕様のディスケットでも使用出来ます。

健康手帳『ディスクインウィークス』自費出版発売中!(2,000円)

P.S. FM←→PCディスクコンバージョンユーティリティ『DISK-MUCH』お求めのユーザーの方々には、手数料6,000円で交換します。オリジナルマスターをご返送下さい。

不思議…?

テープ・ロード・アダプター
テレコとマイコンの間に接続して、波形を成型するので、LOADミスがほとんど、生じなくなります。

OUT IN

価格 8,800円(〒200)

WORLD WIDE COMPUTER SUPER SHOP
COSMOS™ 岡山
〒700 岡山市南方5-6-5 今田ビル2F (県営グランド入口向)
TEL(0862)54-7474 ミニファクス(0862)54-7481
年中無休 AM10:00〜PM7:00
通販は現金書留又は郵便振替 岡山4-12524 コスモス岡山

ユニークなオリジナルソフトで躍進する

CHAMPION SOFT

チャンピオンソフト デザイナー 白木善喜 の異色作シリーズ 好評発売中!!

君もザ・プラネタリウムを見たか

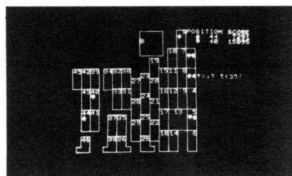
星空の旅

PC8001 MKII
MZ2000, PC8801
パソコンテレビX1
FM-7用

定価 3,500円

- MZ2000用はカラーでも楽しめます。MZ2000用はカラーボードが必要です。カラーの場合はカラーベーシックは不要ですがグラフィックメモリ2枚，カラーディスプレイが必要です。
- パソコンテレビX1用はグラフィックRAMが必要。

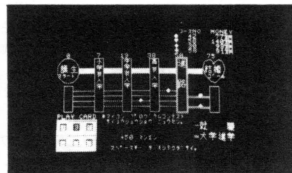
春夏秋冬の四季の星座を満喫できます

シャープパソコンテレビX1, MZ700のユーザーの皆様お待たせいたしました
パソコンテレビX1, MZ700, MZ2000・80B, PC8001, FM-7用好評発売中ソフト

シミュレーションウォーゲーム

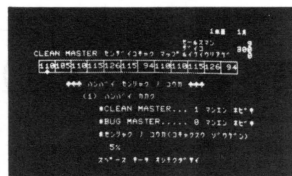
コンバット-47

定価 3,500円

１～４人で楽しめる知的戦略ゲーム。
面白さ抜群。
ルールは単純明解。
47か国のうち君は何か国を占領できるか。

コンピュータダイスプレイゲーム

ウエディングレース

定価 3,500円

１～４人で楽しめるコンピュータダイスプレイゲームの決定版。
めでたく結婚にゴールインできるか。
それとも破産のうきめにあい、地獄行となるか………。

シミュレーションビジネスウォーゲーム

コンピュータウォーズ

定価 3,500円

誰にでもできるシミュレーションビジネスウォーゲームの第１弾。
あなたはコンピュータ会社の営業部長。
現在のシェア10%を３年後に50%にできるか………。

人生シミュレーションゲーム「ザ・サラリーマン」、知的戦略ゲーム第２弾「夢のマイホーム」、第３弾「脱サラ人生」近々発売予定。
1部上場企業で約20年間企画・スタッフ畑を歩んできた人生経験豊富な作者ならではの異色作。　ご期待下さい。

⊙お求めは、お近くのマイコンショップ又は、エヌ・テイ・エル産業㈱宛に品名と機種、氏名、住所、電話番号を記入の上、現金書留にてお送りください。(送料は無料サービス)

発売元 NTL エヌ・テイ・エル産業株式会社
〒657 神戸市灘区篠原北町１丁目５番13号
TEL (078) 881－1914

制作 CS 株式会社チャンピオンソフト
〒530 大阪市北区西天満５丁目15番18号(実業ビル)
TEL (06) 365－9900

EDITOR'S ROOM

☆ＦＭのハード・ソフトに詳しい人、自分でプログラムが書ける人を募集します。「ＦＭのことならまかしとけ！」という人、大歓迎です。性別、年齢、職業などは問いません。我こそは、と思う方、03-261-4095 Ｏｈ！ＦＭ編集部・松岡までお電話をください。

オリジナル・プログラムも募集します。直接お持ちくださるか、郵送してください。郵送の場合は、①内容の説明、②フローチャート、③テープかディスケットを同封し、住所・氏名・年齢・職業を明記のうえ、日本ソフトバンクＯｈ！ＦＭ編集部あて、お送りください。

本誌に掲載された分は、当社規定による原稿料をさしあげます。

宛先

〒102 千代田区四番町３ 四番町ハイツ501

(株)日本ソフトバンク Ｏｈ！ＦＭ編集部

本誌の内容に関して，電話による質問を受け付けます。

☎03-265-5789

質問はバグ情報についてのみ受け付けます。機械の取り扱いなどについてのお問い合わせは、ご遠慮ください。マニュアルを熟読してください。

■読者アンケートの圧倒的多数がせめて隔月刊、できれば月刊化してほしいとのことでした。編集担当が頑張った結果、ようやく隔月刊と決定しました。次号は5月、続いて7、9、11月の18日発売となります。来年こそ月刊に、と担当は言ってますのでよろしく……。（Ｔ）

■原稿をもう少し待って、というのが筆者。原稿を早く入れないと本が出ないよ、というのが印刷所。その板ばさみになるのが編集者だ。この号から本誌担当がMに代わった。印刷所との折衝、筆者との打合せなどテキパキとこなす。さすがこの道10余年のベテランだ。（Ｏ）

■今号から担当者がかわりＦＭにも春がやってきました。新担当者は頼もしいお姉様です。ところで忙しい日々が続くと何か楽しみをつくることにしていますが、今のところは４月２日サラ・ヴォーンのコンサートと動物園巡りが楽しみとなっております。（Ｔ）

■ＦＭの読者の中には、6800、6809ＣＰＵのファンの方が多いと思います。２MHz ＣＰＵの力を発揮するのはこれから、マシンを開発ツールとしてＦＬＥＸ、ＯＳ-９、ＣＰ/Ｍなどをあやつり、昼休みにはゲームマシンに使っている人も多いと聞く。さて君は…？（Ｋ）

■本号より編集スタッフに加わりました。まずはＩＣにたとえて自己紹介。外観は初期の頃の厚膜ＩＣといったところですが、大容量のダイナミックＲＡＭめざしています。書き込み不能、読み出しオンリーのＲＯＭにならないようがんばりますので、よろしく。（Ｍ）

●編集 渡辺妙子 松岡真理

●技術 松田辰夫 小林初雄 徳永 聡

●写真 浜崎 昭 杉山和美

●イラスト 松浦靖志

●表紙イラスト ㈱マジック・バス

●レイアウト 創美レイアウト 堀 立明
レモンデザインルーム

●今号の表紙●

ＪＵＮグラフィックステーション４Ｄを使用

本誌今号の表紙は、㈱ＪＵＮのパーソナルグラフィックスステーション４Ｄを使ったもの。右の原画イラストをもとに、リボン部分をデジタイザで入力後、画像を順次縮小、レインボーカラーにペインティングした。

この４Ｄへの入力は、デジタイザのほか、テレビカメラでもＯＫ。原画そのままの画像に、色のバリエーションをつけたり、移動、回転、ズーミングなど自由自在。

発色は1,600万色の中から自由に256色が選べ、同時に256色を表示可能。また512×512ドットの高解像度で、微妙な色調もダイヤルひとつでコントロールできる。

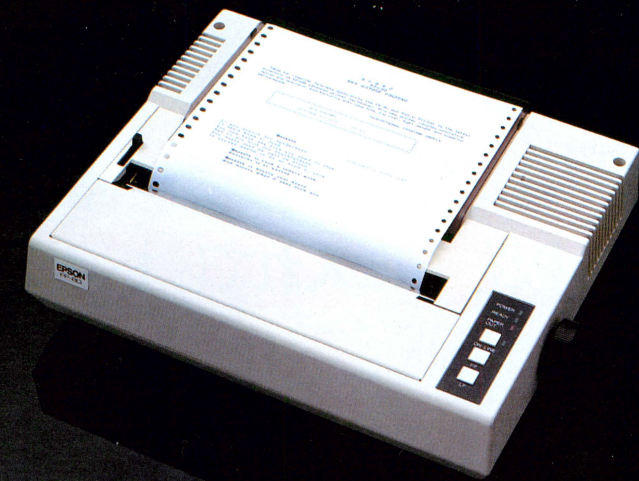
ＣＰＵには16ビットＭＣ68Ｋを使用、64Ｋバイト（512Ｋまで拡張可）のメモリをもっている。プログラミング言語はFORTH系を採用している。

パターンの色や形を自由に変えられるため、服飾・インテリアなどのデザイン分野や、天気図や断層断面図などの科学技術・数育分野、アニメーション分野など、種々のジャンルで応用できる。

なお、４Ｄ本体の価格は230万円。レンタルもある。

●広告索引●

季刊 Oh!FM 第２号（'83春季号）

・昭和58年４月１日発行
・発行人／孫 正義
・編集人／田鎖 洋治郎
・発行元／株式会社日本ソフトバンク
〒102 東京都千代田区四番町２－１
・電話／東京(03)263-3690
編集室直通 (03)261－4095
・大阪営業所／〒555 大阪市西淀川区花川１－16－14
電話／大阪(06)475－0028
・印刷 図書印刷㈱ TEL 453－2550

エプソンの最新鋭。超高性能・経済派プリンタ2機種、新登場。

FP-80

高速印字と多彩な文字バリエーションの強力インテリジェントターミナル ●ファンフォールド紙、ロール紙、レター用紙が使える可変ピンフィード、フリクション＆トラクタフィード(オプション)方式●標準インタフェイス：セントロニクス準拠8ビットパラレル ●インパクト・ドット・マトリックス●超高精度ビットイメージプリンティング9種類●印字速度、160CPS(テキスト時)●アンダーライン機能、スーパー／サブスクリプト文字、強調文字など多彩な文字種に加え、国際文字印字(9ヶ国対応)、ダウンロード文字(ユーザー定義可能文字)、さらにプロポーショナル印字も可能。また、すべての文字種は同一行混在可。左右のマージン設定、水平タブ、垂直タブ、などの機能によりフォーマティングが自由自在。

￥149,800／￥152,800(PC-8001専用)／￥153,800(PC-8801/9801専用)

RP-80

MPの技術が生んだ高性能経済派プリンタ

●4″～10″のファンフォールド紙が使えるトラクタフィード方式●標準インタフェイス：セントロニクス準拠8ビットパラレル●インパクト・ドット・マトリックス●単密度480ドット/行、倍密度、倍速倍密640ドット／行、CRTグラフィックス640、720ドット／行、4倍密度1920ドット／行の高精度ビットイメージプリンティング6種類●印字速度：100CPS(テキスト時)●アンダーライン機能、スーパー／サブスクリプト文字、強調文字など多彩な文字種と豊富なファンクション

￥89,000

DP-20

鮮明な印字と簡単な操作のインパクトプリンタ●ディジーホイール採用により、ワードプロセシング等に最適。●ホイールは600万文字と高寿命。●カセット式ディジーホイールおよびリボンメカニズムの採用により、活字・リボン交換がワンタッチ。●文字/行：132文字(1/10)インチ)、158文字(1/12インチ)、198文字(1/15インチ) ●用紙幅：16.5インチ(420㎜) ●キャリッジ送り：10文字/インチ、12文字/インチ、15文字/インチ●ペーパーフィード方式：両方向フリクションフィード)

￥23,000(シリアル仕様)／￥210,000(パラレル仕様)

信頼のプリンタ、MPシリーズ。

MP-80K

14×18ドット、80桁普及型漢字プリンタ ●1台で3役、漢字(双方向印字)/通常文字(双方向最短距離印字)/ビットイメージ(単方向印字左→右)●4″～10″のファンフォールド紙が使える可変スプロケットフィード方式●JIS第1水準準拠、漢字・記号約4,000字装備

￥189,000/漢字コード対応表￥500

MP-100III

高速136桁ビットイメージプリンタの本格派 ●135字/秒の高速プリンティング●底部からの用紙(ファンフォールド紙)挿入もできる2ウェイ(背面・底面)紙送り方式採用の本格ビジネス仕様●1行816ドットのビットイメージプリンティング

￥228,000

MP-130

136桁ビットイメージプリンタの普及版 ●ファンフォールド紙、ワンシートが使えるフリクション＆トラクタフィード方式●桁数は68桁(拡大)、136桁(標準)、116桁(縮小の拡大)、233桁(縮小)の4種類●単密度816ドット/行、倍密度1632ドット/行の高精度ビットイメージプリンティング●アンダーライン機能、スーパー／サブスクリプト文字、強調文字など多彩な文字種と豊富なファンクション

￥189,800/￥192,800(PC-8001専用)

MP-130K

本格的ビジネスユースに最適の高性能24×24ドット、136桁漢字プリンタ ●ファンフォールド紙、ワンシートでB4サイズまで使える可変スプロケット＆フリクションフィード方式●漢字90桁/行、アルファベット・カナ・記号136桁/行●JIS第1水準準拠約4,000字、JIS第2水準準拠約7,000字の漢字ROMを内蔵、JISコードでハンドリングが容易●単密度1224ドット/行、倍密度2448ドット/行の高密度ビットイメージプリンティング

￥510,000(JIS第1水準漢字ROM内蔵)
￥550,000(JIS第2水準漢字ROM内蔵)

各種パソコン対応表

	RP-80	FP-80 標準機	FP-80 PC-8001用	FP-80 PC-8801 /9801用	MP-100 III 標準機	MP-100 III PC-8001用	MP-100 III PC-8801用	MP-100 III M20/23用	MP-130	MP-80K	MP-130K	DP-20
EPSON QC-20	◎●	○●	◎●	◎●	○●	—	—	—	○●	○●	◎●	○
HC-20	◎	○	○	○	○	—	—	—	○	—	—	○
NEC PC-6001	◎●	○	○	○	○	—	—	—	○	○	○	○
PC-8001	◎⊙		◎⊙			◎⊙		—	○	○	○	○
PC-8801	◎●			◎●			◎●	—	○	○	○	○
PC-9801	◎●			◎●		—	◎●	—	○	○	○	○
三菱マルチ16	○●	○●	○●	○●	○●	—	—	—	○	○	○	○
タンディTRS-80	○	○	—	—	○	—	—	—	○	○	○	○
ソードM20/23	○	○	—	—	—	—	—	◎	○	○	○	○
アップルAPPLE II	○	◎⊙	—	—	◎⊙	—	—	—	○	○	○	○
東芝パソピア	◎	◎	◎	—	◎	◎	—	—	○	○	○	○
日立レベルIII	○	○	—	—	—	—	—	—	○	—	—	○

注) HC-20用ROMによりMP-100IIIはコンパチブルになります。
○グラフィックキャラクターが不一致 ◎グラフィックキャラクターが一致 ●ハード・コピーがとれます。⊙オプションによりハードコピーがとれます。

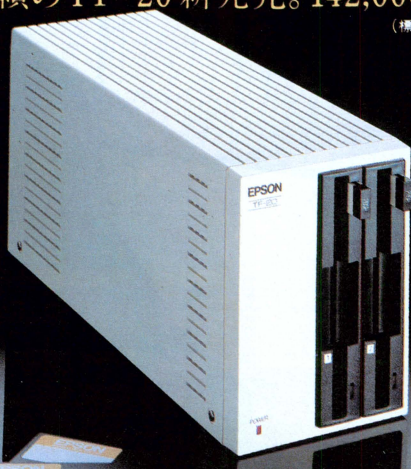
新時代のスリムなフロッピーディスクドライブ装置

(ミニフロッピー。両面倍密度322K×2デッキ内蔵)

TF-20

あなたのパソコンのシステムアップにいま画期的なターミナルフロッピーTF-20の登場です。エプソンのメカトロニクス技術が創り上げた高性能、薄型、信頼のメカニズム。くわえて徹底した低価格を実現しました。いまこそ、あなたのパソコンの能力を大いに引きだして下さい。TF-20はあなたのパソコンライフを一層大きく広げます。

〈TF-20の特長〉**薄型・軽量・コンパクト**：幅が従来品の1/3の新開発薄型ミニフロッピーディスクドライブ2基を搭載、2ドライブのミニフロッピーディスクユニットとしては驚異的な小型化を実現しました。パソコンが小さくなってもフロッピーが…という悩みを解決したスペースセーフぶりです。●**高い操作性**：ディスクを挿入しプッシュボタンを押してセット完了。ディスクを取り出すには、押し込まれているプッシュボタンを更に押すと約1cmディスクが取り出し口よりポップアップ。簡単でしかも確実な操作を可能にしています。●**ハイコストパフォーマンス**：便利なのはわかっているけれど価格が高すぎて…。TF-20は自社開発ディスクドライブを採用し、高い信頼性、性能を確保しながらトータルコストダウンに成功しました。

〈仕様〉●インタフェイス：シュガートSA400コンパチブル●インタフェイスコネクタ：FRC2-C34(DDK)●記録方式：FM、MFM●トラック間セクタ時間：15ms●トラック密度：48TDI●トラック数：80トラック/両面1ドライブ●最大記憶容量：1Mバイト(アンフォーマット)●電源：AC100V(50/60Hz)●消費電力：40W以下●外形寸法：120W×350D×165H(㎜)●重量：7.0kg●価格：エプソンHC-20専用機￥177,000、NEC PC8001・8801専用機￥166,000、富士通MICRO-8専用機￥163,000、

(パソコンもいよいよディスク・ユース時代へ。ターミナルフロッピー、いまエプソンから。)

エプソン株式会社 本社／〒399-07 長野県塩尻市広丘原新田80 ☎02635-4-0271 ●詳しい資料をご希望の方は、はがきに住所、氏名、年令、職業、製品名をお書きの上、もよりの営業所までお申込みください。●東京支店／〒160 東京都新宿区西新宿2-4-1 新宿NSビル内私書箱6026号 ☎03-348-6801・大阪支店／〒530 大阪市北区西天満3-14-16 ☎06-365-5071・名古屋営業所／〒460 名古屋市中区金山1-12-14 ☎052-331-5486・札幌営業所／〒060 札幌市中央区北1条西2丁目札幌時計台ビル6F ☎011-222-2821・仙台営業所／〒980 仙台市本町1-12-12 ☎0222-63-3691・広島営業所／〒730 広島市中区富士見町8-22 ☎082-247-1685・福岡営業所／〒812 福岡市博多区博多駅前3-4-8 ☎092-471-0761

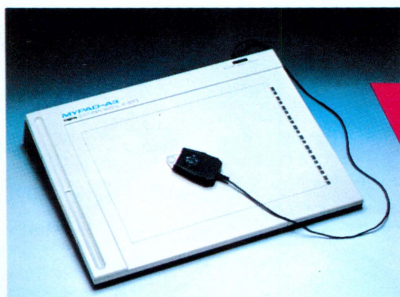
資料請求券 OH／FM

FM流の使い手のすごい奴が現われた。

彼のむかう所、敵なし!!

ついにパソコンの2大道場の看板を手に入れてしまった。

強力パソコン+デジタイザ+ソフト"色・色"
三位一体で冴える技。

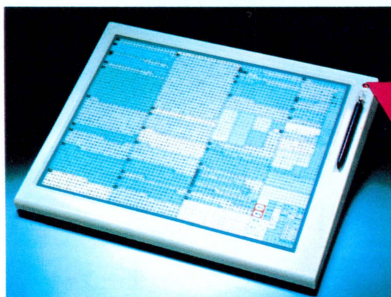
絵や図形、地図などを簡単に、早く入力できます。カーソル又はスタイラスで入力したい図をなぞるだけで入力できますから、もう座標を気にする事は不要です。読取モードも、一点ずつ読取るポイントモード、連続的に(サンプリングレート選択可)読取るストリームモード、連続的に座標データを読取るスイッチストリームモードの3つあります。読取り有効エリアは260×380㎜。小型・安価のデジタイザです。

Logitec

デジタイザ K-510

デジタイザ1台お買上げの方に、トレーナープレゼント!

¥148,000

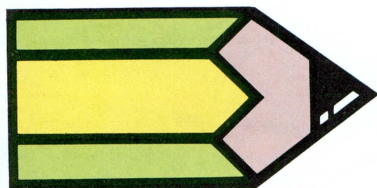
強力パソコン+漢字入力タブレット+ソフト"文々"
三位一体で冴える技。

パーソナルコンピュータに簡単に、しかも素早く漢字等を入力できます。JIS第一水準の漢字・非漢字3816文字が見やすく配置されていますので、入力したい漢字をすぐに探し出せます。そして付属のタッチペンで押すだけで入力できます。簡易入力、安価の漢字入力タブレットです。

Logitec

漢字入力タブレット K-506

¥98,000

これでデジタイザを恐れる必要なし!

ビギナーの方々でもデジタイザを自在に使っていただけます。デジタイザK-510を使って、直線・自由曲線・BOX・BOXFULL・及び色指定(8色)を使って図面を作成できます。また、X-Yプロッタ上へのハードコピーができます。フロッピィを使って図面の登録、呼び出し、移動、縮小、拡大、重ね合せた図面の再登録ができます。ファイル概念のX-Yプロッタ、フロピィユニットへの展開ができます。

Mr.Brain
サンプルプログラムシリーズ
GRAPHIC
えんぴつ "色・色"(いろ いろ)

オフィスでのパソコン活用第一歩。

めんどうだった和文タイプの機能を簡便にまとめました。パソコンにこれから入門するという方々にも最適ですが、もちろん、パソコンのかなりのベテランの方々にも必要な機能を使い易くまとめたこのソフトウェアの使い勝手の良さは、ご満足をあたえるものと思います。

日本語ワードプロセッサ"文々"シリーズ
Mr.Brain.

"文々"…¥13,500 "文々II"…¥18,800

先端技術が生きているパソコン!

●まずご家庭のカラーテレビに接続して楽しめます。●F-BASICを標準実装し、盛り沢山の機能をサポート。●ドット毎に8色まで色指定できるグラフィック機能を内蔵●サウンドクルージングが楽しめるサウンド機能を内蔵●読みやすく使いやすい日本語標示機能●入門用から実務まで1台でOK ¥126,000

FUJITSU
FM-7

16ビットCPUを生かしきったパソコン!

●ミニフロッピィディスク標準実装で、分離型 ●日本語処理機能をいっそう充実 ●豊富なOSが使えます。●機能強化したF-BASIC ●多彩な機能を発揮するCRT表示●目的に応じて選択できる周辺装置 ●メモリは128KBを標準実装、最大1MB。

FM-11〈EX〉…¥398,000
FM-11〈AD〉…¥338,000
FM-11〈ST〉…¥268,000

FUJITSU
FM-11

関東電子株式会社 POWERFUL & CLEVER
マイコン営業本部〒101東京都千代田区神田須田町2-23-1芝崎ビル☎03-251-1101(代)
大阪支店〒556大阪市浪速区日本橋3-3-5カトウビル☎06-632-0207(代)

●札幌営業所☎011-832-0131 ●仙台営業所☎0222-33-0257 ●長岡営業所☎0258-32-8888●群馬営業所☎0270-23-2301
●多摩営業所☎0424-65-8188●町田営業所☎0427-28-8882●千葉営業所☎0472-48-2955●名古屋営業所☎052-263-1693
●京都営業所☎075-343-0995 ●広島営業所☎082-227-5536 ●福岡営業所☎092-474-5777●熊本営業所☎0963-26-1166

ソフトバンクはソフトの流通化新時代を築いていきます。

あなたに大接近。

ひき続き 加盟店募集中！

特別加盟店

現在市販されているソフトのほとんどを揃えています。

■**マイコンショップCSK(西口)**
〒160 新宿区西新宿1-12-18(勧業角丸証券隣)
03-342-1901〈担当・前田店長〉

■**西武百貨店池袋店マイコンショップ**
〒171 豊島区南池袋1-28-1
03-981-0111 内線2928〈担当・池田、鈴木〉

■**ラオックス新宿店マイコンシティ**
〒160 新宿区新宿3-15-16 ラオックスビル5F
03-350-1241〈担当・菅原店長〉

■**コンピュータハウスヤマギワテクニカ**
〒101 千代田区外神田4-3-1
03-253-0121(代)〈担当・原〉

■**大塚OAセンター八重州**
〒104 中央区八重州2-1-5 東京駅前ビル2
03-281-4711〈担当・金子〉

■**J&Pテクノランド**
〒556 大阪市浪速区日本橋5-6-7
06-644-1413〈担当・勝田店長〉

■**J&Pメディアランド**
〒556 大阪市浪速区日本橋4-11-3
06-644-1613〈担当・山本店長〉

■**マイコンショップCSK**
〒530 大阪市北区梅田1-1-3 大阪駅前第3ビルB-1
06-345-3351〈担当・四元〉

■**YDKシステムセンター**
〒160 新宿区西新宿2-4-1 新宿NSビル1F
03-342-2441〈担当・宮本店長〉

■**第一家電OAセンター**
〒101 千代田区神田佐久間町1-15
03-253-0808〈担当・飯田〉

ソフトバンク加盟店なら、あなたの欲しいソフトのほとんどが入手できます。お店の方にご相談ください。

ソフトバンクにご連絡下さい。あなたのお近くのソフトバンク加盟店をご紹介します。

株式会社 **日本ソフトバンク** SOFT BANK

〒102 東京都千代田区四番町2-1 03(263)3690　大阪営業所 〒555 大阪市西淀川区花川1-16-14 06(475)0028

コンピュータショ

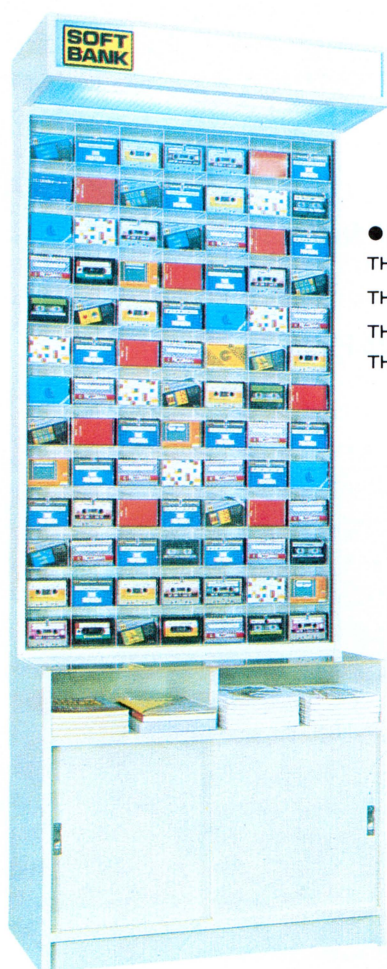
あのソフトバンクのスタンダ

パソコン大好き人間はソフトウェアに飢えています。
彼らは口々に叫ぶのです「もっとソフトを!!」

今はソフトの時代! ソフト需要の急激な増加の多様化にあなたのショップは対応できますか?

今どんなソフトが人気があるのか? どんなソフトをどれだけ揃えたらいいのか?……コンピュータショップの経営者にとっては頭の痛い問題に違いありません。

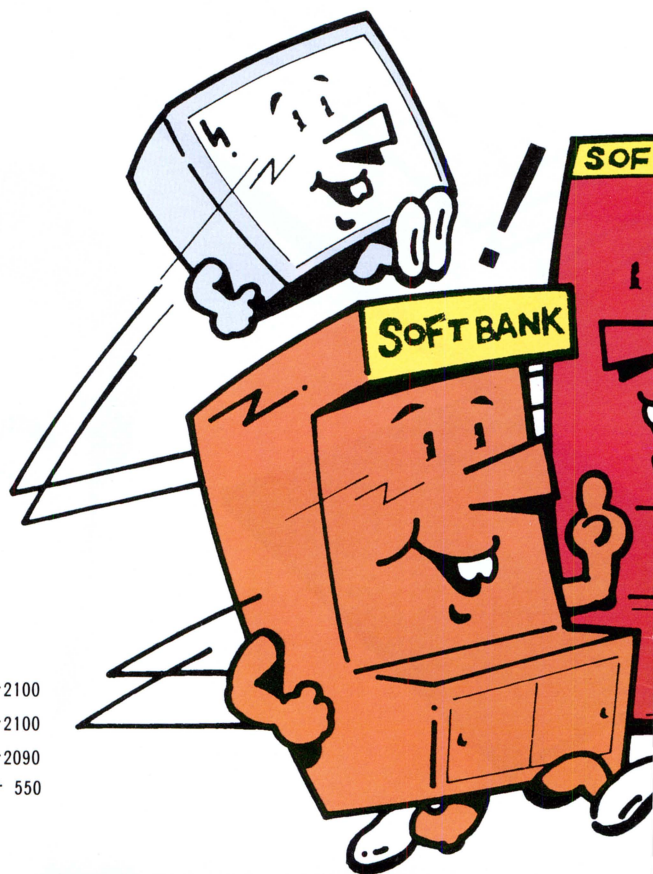
THE455

ゲーム用　91種類　各5本収納可　　定価105,000円
(横900　奥450　縦2100)
ビジネス、ブックタイプ用3段　各20本収納可　定価105,000円
(横900　奥500　縦2100)

●スタンダードキャビネットケース／寸法

THE455（ゲーム用）	ヨコ900 オク450 タテ2100
THE455（ブックタイプ用 ビジネス用）	ヨコ900 オク500 タテ2100
THE375（ゲーム用）	ヨコ625 オク450 タテ2090
THE125（ゲーム用）	ヨコ580 オク240 タテ 550

●スタンダードキャビネットケース／価格

THE455（ゲーム用）	定価105,000円
THE455（ブックタイプ用 ビジネス用）	定価105,000円
THE375（ゲーム用）	定価 52,000円
THE125（ゲーム用）	定価 8,900円

SOFT BANK

ップのみなさんへ

ードキャビネットが勢揃い！

そこでソフトバンクはこう提案をします。

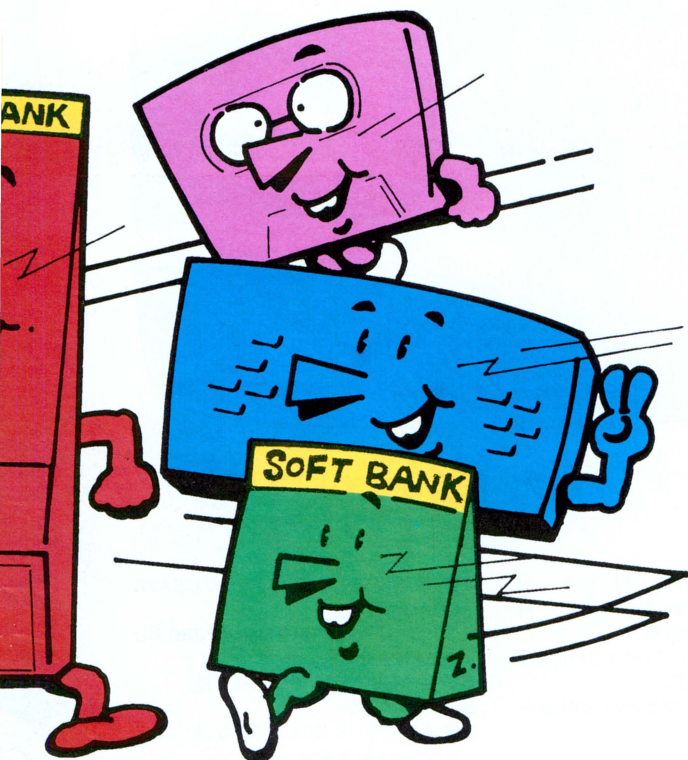
ソフトバンクでは、機種別、人気ソフト、新作ソフトのヒットチャートを随時打ち出しており、いつでも新鮮なソフトをキャビネットに納める事ができます。このキャビネットは、そのまま店内にディスプレイできるようになっています。その上、ディスプレイの下部スペースは、補充・ストック用・売れ筋ソフトを効率よく収納できる設計になっています。

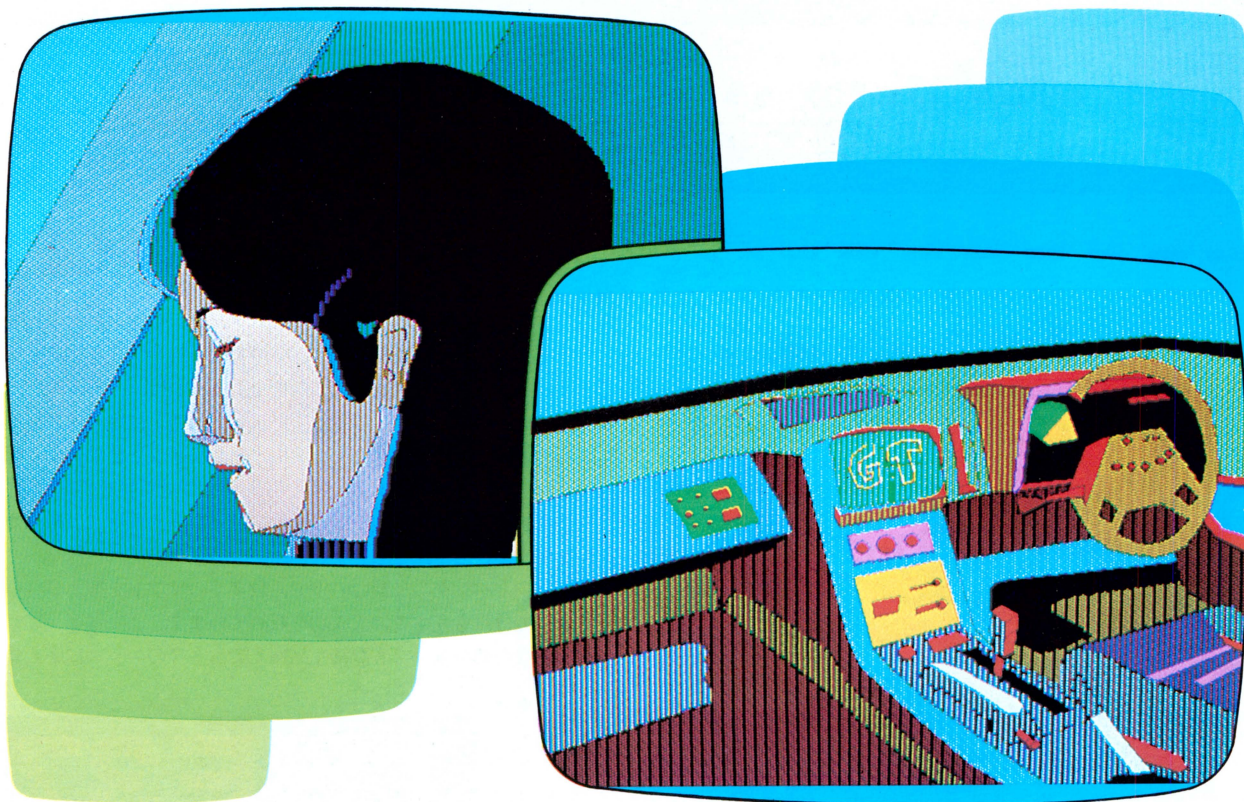
これからのコンピュータショップは、数多くのソフトを揃える事が第一条件です。

ソフトバンクの加盟店の中で、数多くのソフトを揃えているショップでは、もうすでに大成功を収めています。

あなたのショップも豊富な品揃えでお客様のニーズに応えませんか。

THE375

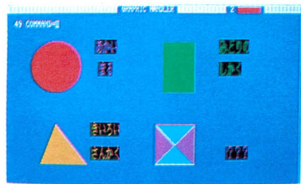
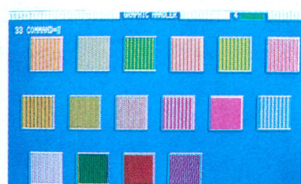
ゲーム用　75種類　各５本収納可　　定価52,000円
（横625　奥450　縦2090）

THE125

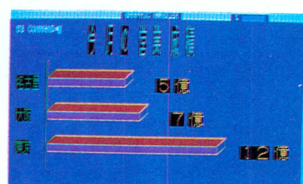
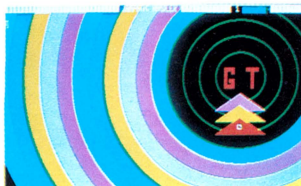
ゲーム用　25種類　各５本収納可　　定価8,900円
（横580　奥240　縦550）

THE GRAPHIC TOOL
グラフィック用簡易言語GT

グラフィック自由自在。

- **コマンド一発**で直線、円、ダ円、色塗り、四角、メッシュ、点線が描ける他グラフィック用の強力コマンドを多数装備。
- 今まで数時間を費やしていたカラーの画面デザインが**数分で完了！**
- しかも簡易言語だから**プログラムの知識がなくてもOK！**
- 作った画面データはディスク、テープに**SAVE、LOADが可能。**
- 赤、青、緑色の3原色を**1000通りの組み合せでまぜ合せ可能。**
- プリンターによる**ハードコピーも可能。**
- トレスモード有。
- 驚異のKコマンド、漢字、ひらがな、記号の色指定サイズ指定可能。

作成したグラフィック画面がBASIC文になる

BASICジェネレータ機能装備。

●教育用グラフィックデザインプログラムとして

●グラフィックデザイナーの配色シュミレーションとして

●アプリケーションプログラムのメッセージ画面設計ツールとして

●ビジュアルな図形資料として会議用にレポート提出

●デザインコンペの記録保存プレゼン用に

●簡易ポスターデザインのツールとして

日本SEグループ
ニチコン
㈱日本コンピュータ設計
〒150 東京都渋谷区渋谷2-22-7 共栄ビル
TEL.(03)498-2616

Oh!FM ㈱日本ソフトバンク発行 Printed in Japan 定価480円 雑誌02199-04